

The image shows a large grid of symbols arranged in a pattern. The symbols are grouped into four main vertical columns. The first column contains 'SSS' repeated seven times at the top, followed by 'SSSSSSSS' repeated six times. The second column contains 'YYY' repeated five times at the top, followed by 'YYYY' repeated three times. The third column contains 'YYY' repeated five times at the top, followed by 'SSS' repeated five times. The fourth column contains 'SSSSSSSS' repeated six times at the top, followed by 'SSSSSSSS' repeated five times. The symbols are rendered in a bold, black, sans-serif font.

PPPPPPPP	AAAAAA	GGGGGGGG	EEEEEEEEE	FFFFFFFFF	AAAAAA	UU	UU	LL	TTTTTTTTTT
PPPPPPPP	AAAAAA	GGGGGGGG	EEEEEEEEE	FFFFFFFFF	AAAAAA	UU	UU	LL	TT
PP PP	AA AA	GG	EE	FF	AA	AA	UU	UU	TT
PP PP	AA AA	GG	EE	FF	AA	AA	UU	UU	TT
PP PP	AA AA	GG	EE	FF	AA	AA	UU	UU	TT
PPPPPPPP	AA AA	GG	EEEEEEEEE	FFFFFFFFF	AA	AA	UU	UU	TT
PPPPPPPP	AA AA	GG	EEEEEEEEE	FFFFFFFFF	AA	AA	UU	UU	TT
PP	AAAAAAAAAA	GG	GGGGGG	EE	FF	AAAAAAA	UU	UU	TT
PP	AAAAAAAAAA	GG	GGGGGG	EE	FF	AAAAAAA	UU	UU	TT
PP	AA AA	GG	GG	EE	FF	AA	AA	UU	TT
PP	AA AA	GG	GG	EE	FF	AA	AA	UU	TT
PP	AA AA	GGGGGG	EEEEEEEEE	FF	AA	AA	UUUUUUUUUU	LLLLLLLLLL	TT
PP	AA AA	GGGGGG	EEEEEEEEE	FF	AA	AA	UUUUUUUUUU	LLLLLLLLLL	TT

LL	IIIIII	SSSSSSSS
LL	IIIIII	SSSSSSSS
LL	II	SS
LL	II	SS
LL	II	SS
LL	II	SSSSSS
LL	II	SSSSSS
LL	II	SS
LL	II	SS
LL	II	SS
LLLLLLLL	IIIIII	SSSSSSSS
LLLLLLLL	IIIIII	SSSSSSSS

(1)	36	HISTORY : DETAILED
(2)	60	DECLARATIONS
(3)	165	PAGE FAULT HANDLER
(4)	221	SYSTEM PAGE FAULT, ESTABLISH PAGE TYPE
(5)	298	EXCEPTION ENTRY POINT - PAGE TYPE DISPATCHER
(6)	402	PAGE FILE, SECTION TABLE INDEX, OR GLOBAL PAGE
(7)	473	PAGE NOT RESIDENT, QUEUE A READ REQUEST
(8)	828	FORM A CLUSTER OF PAGES TO READ
(9)	990	DEMAND ZERO PAGE
(10)	1106	FREE, MODIFIED, OR BAD PAGE LIST, RELEASE PENDING
(11)	1254	SCANDEADPT - SCAN A DEAD PAGE TABLE FOR TRANSITION PAGES
(12)	1355	WSLEPFN - FETCH PFN FROM WORKING SET LIST ENTRY
(13)	1406	FREWSLE - FREE A WORKING SET LIST ENTRY
(15)	1720	DELWSLEX - DELETE WORKING SET LIST ENTRY BY INDEX
(17)	1802	ININEWPFN - ALLOCATE AND INIT A NEW PFN
(19)	1869	MAKEWSLE - MAKE A WORKING SET LIST ENTRY
(21)	1967	LOCKPGTB - LOCK PAGE TABLE
(22)	2020	INCPTREF - INCREMENT PAGE TABLE REFERENCE COUNT
(24)	2098	DECPTREF - DECREMENT PAGE TABLE REFERENCE COUNT
(26)	2178	DECPHDREF - DECREMENT PROCESS HEADER REFERENCE COUNT
(27)	2244	INIBLDPKT - INIT FOR CALLING BUILDPKT

0000 1 .TITLE PAGEFAULT - TRANSLATION NOT VALID EXCEPTION HANDLER  
0000 2 :IDENT 'V04-000'  
0000 3 :\*\*\*\*\*  
0000 4 :  
0000 5 :\*  
0000 6 :\* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY  
0000 7 :\* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.  
0000 8 :\* ALL RIGHTS RESERVED.  
0000 9 :\*  
0000 10 :\* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
0000 11 :\* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
0000 12 :\* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
0000 13 :\* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
0000 14 :\* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
0000 15 :\* TRANSFERRED.  
0000 16 :\*  
0000 17 :\* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
0000 18 :\* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
0000 19 :\* CORPORATION.  
0000 20 :\*  
0000 21 :\* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
0000 22 :\* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.  
0000 23 :\*  
0000 24 :\*  
0000 25 :\*\*\*\*\*  
0000 26 :  
0000 27 :++  
0000 28 :FACILITY: EXECUTIVE, TRANSLATION NOT VALID EXCEPTION HANDLER  
0000 29 :  
0000 30 :ABSTRACT:  
0000 31 :  
0000 32 :ENVIRONMENT:  
0000 33 :  
0000 34 :--  
0000 35 :  
0000 36 :.SBTTL HISTORY ; DETAILED  
0000 37 :  
0000 38 :AUTHOR: PETER H. LIPMAN , CREATION DATE: 14-SEP-76  
0000 39 :  
0000 40 :MODIFIED BY:  
0000 41 :  
0000 42 :V03-008 WMC0002 Wayne Cardoza 24-Jul-1984  
0000 43 :Comparison on modified page list waitlimit should be CMPL.  
0000 44 :  
0000 45 :V03-007 WMC0001 Wayne Cardoza 05-MAY-1983  
0000 46 :Fix trigger for dead page table scan to account for  
0000 47 :locked page tables in dynamic part of working set.  
0000 48 :  
0000 49 :V03-006 TCM0001 Trudy C. Matthews 31-Mar-1983  
0000 50 :Change references to working set fields in PHD so that  
0000 51 :they are used as unsigned words.  
0000 52 :  
0000 53 :V03-005 KDM54836 Kathleen D. Morse 16-Mar-1983  
0000 54 :Add deadlock detection: do not allow process holding  
0000 55 :a mutex to wait for swap file space.  
0000 56 :  
0000 57 :V03-004 KDM0002 Kathleen D. Morse 28-Jun-1982

PAGEFAULT  
V04-000

- TRANSLATION NOT VALID EXCEPTION HISTORY ; DETAILED F 5 HANDLE 16-SEP-1984 00:43:02 VAX/VMS Macro V04-00  
5-SEP-1984 03:45:49 [SYS.SRC]PAGEFAULT.MAR;1

Page 2  
(1)

0000 58 :

Added \$PRTDEF.

```
0000 60 .SBTTL DECLARATIONS
0000 61 :
0000 62 : INCLUDE FILES:
0000 63 :
0000 64 $CADEF ;CONDITIONAL ASSEMBLY DEFINITIONS
0000 65 $IPLDEF ;PROCESSOR PRIORITY LEVEL DEFINITIONS
0000 66 $IRPDEF ;I/O REQUEST PACKET DEFINITIONS
0000 67 $PCBDEF ;PROCESS CONTROL BLOCK DEFINITIONS
0000 68 $PHDDEF ;PROCESS HEADER DEFINTIONS
0000 69 $PRDEF ;PROCESSOR REGISTER DEFINITIONS
0000 70 $PFLDEF ;PAGE FILE CONTROL BLOCK DEFINITIONS
0000 71 $PFNDEF ;PFN DATA BASE DEFINITIONS
0000 72 $PRTDDEF ;PROTECTION FIELD DEFINITIONS
0000 73 $PSLDEF ;PROCESSOR STATUS LONG WORD DEFINITIONS
0000 74 $PTEDEF ;PAGE TABLE ENTRY OFFSETS
0000 75 $RSNDEF ;RESOURCE NAME DEFINITIONS
0000 76 $SECDEF ;SECTION TABLE DEFINITIONS
0000 77 $SSDEF ;SYSTEM STATUS DEFINITIONS
0000 78 $VADEF ;VIRTUAL ADDRESS VIELDS
0000 79 $WQHDEF ;WAIT QUEUE HEADER DEFINITIONS
0000 80 $WSLDEF ;WORKING SET LIST DEFINITIONS
0000 81 :
0000 82 : EXTERNAL SYMBOLS:
0000 83 :
0000 84 :
0000 85 :
0000 86 : MACROS:
0000 87 :
0000 88 :
0000 89 :
0000 90 : EQUATED SYMBOLS:
0000 91 :
0000 92 :
0000 93 $OFFSET <4*6>,POSITIVE,<-
0000 94 FLTCTL,- ;OFFSET TO FAULT CONTROL BITS
0000 95 FLTVA,- ;OFFSET TO FAULT VIRTUAL ADDRESS
0000 96 FLTPC,- ;OFFSET TO FAULT PC
0000 97 FLTPSL,- ;OFFSET TO FAULT PSL
0000 98 >
0018 FLTCTL:
001C FLTVA:
0020 FLTPC:
0024 FLTPSL:
0000 99 :
0000 100 $VIELD PGF,0,<-
0000 101 LENVIO,- ;DEFINE PAGE FAULT CONTROL BITS
0000 102 PGTBFLT,- ;LENGTH VIOLATION
0000 103 WRTACC- ;PAGE TABLE FAULT
0000 104 > ;REFERENCE WAS WRITE OR MODIFY
0000 105 :
0000 106 : OFFSETS INTO I/O PACKET WHILE BEING USED AS SCRATCH STORAGE FOR CLUSTERING
0000 107 :
0000 108 $OFFSET 0,POSITIVE,<-
0000 109 PIEDAT,- ;MASTER PTE CONTENTS
0000 110 SVAPTE,- ;MASTER PTE ADDRESS
0000 111 SIZE_TYPE,- ;RESERVED FOR SIZE AND TYPE
0000 112 VA,- ;VIRTUAL ADDRESS
```

0000	113	AST,-	: AST INFO
0000	114	ASTPRM,-	: AND PARAMETER
0000	115	GPTX,-	: PROCESS PTE CONTENT FOR GLOBAL PAGE
0000	116	GPTX_PTE,-	: PROCESS PTE ADR FOR GLOBAL PAGE
0000	117	<CLUSTER,1>,-	: DESIRED CLUSTER SIZE
0000	118	<COUNT,1>,-	: CURRENT COUNT OF PAGES
0000	119	<STATE,1>,-	: SAVED PFN STATE BYTE
0000	120	<PRI,1>,-	: PRIORITY OF I/O TRANSFER
0000	121	BAK,-	: PFN BACKING STORE ADDRESS
0000	122	INC1,-	: + OR - 1
0000	123	INC4,-	: + OR - 4
0000	124	INC512,-	: + OR - 512
0000	125	VBN,-	: VIRTUAL BLOCK NUMBER
0000	126	WINDOW,-	: WINDOW CONTROL BLOCK ADR
0000	127	FP_SAV,-	: SAVED FP
0000	128	<PCB_SAV,B>,-	: SAVED PCB, PHD ADDRESS
0000	129	PHVREFCADR,-	: PROCESS HEADER REFERENCE COUNT ADDRESS
0000	130	<CLU_SCRATCH_SIZ,0>-	: SIZE OF THIS SCRATCH AREA
0000	131	>	
0000		PTEDAT:	
0004		SVAPTE:	
0008		SIZE_TYPE:	
000C		VA:	
0010		AST:	
0014		ASTPRM:	
0018		GPTX:	
001C		GPTX_PTE:	
0020		CLUSTER:	
0021		COUNT:	
0022		STATE:	
0023		PRI:	
0024		BAK:	
0028		INC1:	
002C		INC4:	
0030		INC512:	
0034		VBN:	
0038		WINDOW:	
003C		FP_SAV:	
0040		PCB_SAV:	
0048		PHVREFCADR:	
004C		CLU_SCRATCH_SIZ:	
0000	132		
0000	133	ASSUME CLU_SCRATCH_SIZ LE IRP\$C_LENGTH	
0000	134	ASSUME SIZE_TYPE EQ IRP\$W_SIZE	
0000	135	ASSUME AST EQ IRP\$L_AST	
0000	136	ASSUME ASTPRM EQ IRP\$L_ASTPRM	
0000	137	ASSUME PRI EQ IRPSB_PRI	
0000	138	:	
0000	139	OWN STORAGE:	
0000	140	:	
00000000	141	.PSECT \$SS210, LONG	
0000	142	:	
0000	143	STATISTICS	
0000	144	:	
00000004	145	PMSSGL_FAULTS::	: COUNT OF PAGE FAULTS
0000	146	.BLKL 1	: OBSOLETE NAME
0004	147	PMSSGL_RDFLTS::	

00000008	0004	148 PMSSGL_PREADS::	
	0008	149 .BLKL 1	;PAGE READS
0000000C	0008	150 PMSSGL_PREADIO::	
	000C	151 .BLKL 1	;I/O REQUESTS TO READ THE PAGES
00000010	000C	152 PMSSGL_PWRITES::	
	0010	153 .BLKL 1	;MODIFIED PAGES WRITTEN
00000014	0010	154 PMSSGL_PWRITIO::	
	0014	155 .BLKL 1	;I/O REQUESTS TO WRITE THE MODIFIED PAGES
00000018	0014	156 PMSSGL_DZROFLTS::	
	0018	157 .BLKL 1	;DEMAND ZERO PAGE FAULTS
00000038	0018	158 PMSSAL_TRANSFLT::	
	0038	159 .BLKL <1@PFN\$\$_LOC>	;PAGE FAULTS OUT OF TRANSITION STATES
0000003C	0038	160 PMSSGL_DPTSCN::	
	003C	161 .BLKL 1	;DEAD PAGE TABLE SCANS
00000040	003C	162 PMSSGL_GVALID::	
	163	.BLKL 1	;GLOBAL VALID FAULTS

0040 165 .SBTTL PAGE FAULT HANDLER  
0040 166  
0040 167 :++  
0040 168 : FUNCTIONAL DESCRIPTION:  
0040 169 :  
0040 170 : THIS MODULE CONTAINS THE PAGEFAULT HANDLER. IT IS ENTERED VIA A  
0040 171 : TRANSLATION-NOT-VALID FAULT. AT THE TIME OF A FAULT, THE KERNEL  
0040 172 : STACK CONTAINS THE FOLLOWING INFORMATION:  
0040 173 :  
0040 174 :  
0040 175 : REASON MASK :--> BIT 0 - ALWAYS 0 FOR  
0040 176 : TRANS-NOT-VALID FAULTS  
0040 177 : INVALID VIRTUAL ADDRESS : BIT 1 - 0 INDIC VIRT ADR NOT VALID  
0040 178 : PC OF FAULTING INSTRUCTION : 1 INDIC ASSOC PTE NOT VALID  
0040 179 :  
0040 180 :  
0040 181 : PSL OF FAULTING INSTRUCTION : BIT 2 - 0 INDIC READ ACCESS  
0040 182 :  
0040 183 :  
0040 184 : CALLING SEQUENCE:  
0040 185 : NONE  
0040 186 :  
0040 187 :  
0040 188 : INPUT PARAMETERS:  
0040 189 : NONE  
0040 190 :  
0040 191 : IMPLICIT INPUTS:  
0040 192 : NONE  
0040 193 :  
0040 194 : OUTPUT PARAMETERS:  
0040 195 : NONE  
0040 196 :  
0040 197 : IMPLICIT OUTPUTS:  
0040 198 : NONE  
0040 199 :  
0040 200 : COMPLETION CODES:  
0040 201 : NONE  
0040 202 :  
0040 203 : SIDE EFFECTS:  
0040 204 : NONE  
0040 205 :  
0040 206 :--  
0040 207 :  
0040 208 :  
0040 209 :\*\*\*\*\*  
0040 210 :\*\*\*\*\*  
0040 211 :\*\*\*\*\* THIS ENTIRE MODULE MUST BE RESIDENT \*\*\*\*\*  
0040 212 :\*\*\*\*\*  
00000000 213 .PSECT \$MMG\$COD, LONG  
0000 214 :\*\*\*\*\*  
0000 215 :\*\*\*\*\*  
0000 216 :  
0000 217 : LIST MEB  
0000 218 :  
FEFF 0000 219 IPLHI: BUG\_CHECK PGFIPLHI,FATAL ;IPL TOO HIGH FOR PAGE FAULT  
0004' 0002 0000 .WORD ^XFEFF  
0000 :IIF IDN <FATAL>,<FATAL> , .WORD BUG\$\_PGFIPLHI!4

```

0004 221 .SBTTL SYSTEM PAGE FAULT, ESTABLISH PAGE TYPE
0004 222
0004 223
0004 224 :
0004 225 : BAD SYSTEM PAGE - PROCESS HEADER OR PAGE TABLE PAGE FOR ANOTHER PROCESS
0004 226 :
0004 227 : IF THE PROCESS HEADER HAS JUST BEEN INSWAPPED (PHDSV_NOACCVIO IS SET),
0004 228 : SIMPLY DISMISS THE FAULT. IN ALL OTHER CASES, REPORT AN ACCESS VIOLATION
0004 229 :
0004 230 :
0004 231 : IF THE PROCESS WAS OUTSWAPPED WHILE ACCESSING ITS OWN HEADER,
0004 232 : DISMISSING THE EXCEPTION WILL CAUSE THE REFERENCE TO OCCUR AGAIN,
0004 233 : BUT THIS TIME TO THE CORRECT BALANCE SLOT.
0004 234 :
0004 235 : IF THE PROCESS WAS MAKING AN ILLEGAL REFERENCE TO THE HEADER OF ANOTHER
0004 236 : PROCESS, DISMISSING THE EXCEPTION WILL CAUSE THE SAME ILLEGAL REFERENCE
0004 237 : TO OCCUR AGAIN, BUT NOW WITH PHDSV_ACCVIO CLEAR, CAUSING AN ACCESS
0004 238 : VIOLATION TO BE REPORTED.
0004 239 :
0004 240 BADSYSPAG:
03 36 A5 03 E4 0004 241 BBSC #PHDSV_NOACCVIO,PHD$W_FLAGS(R5),10$ ;BRANCH IF HEADER JUST INSWAPPED
0687 31 0009 242 BRW ACVIOLAT ;FAKE AN ACCESS VIOLATION
05F7 31 000C 243 10$: BRW PGFCOMPLETE ;SIMPLY DISMISS THE PAGE FAULT
000F 244
000F 245
000F 246 :
000F 247 : SEE IF PAGE IS A GLOBAL PAGE TABLE PAGE, OTHERWISE ERROR
000F 248 :
000F 249 GPGTBL:
00000000'EF 50 B1 000F 250 CMPW R0,SGN$GL_BALSETCT ;SYSTEM BALANCE SET SLOT?
EC 19 0016 251 BLSS BADSYSPAG ;BRANCH IF NOT
0000'CF 52 D1 0018 252 CMPL R2,W$MMG$GL_MAXGpte ;LEGAL GPTE ADDRESS
52 E5 1E 001D 253 BGEQU BADSYSPAG
52 OA 90 001F 254 MOVB #WSL$C_GPGTBL,R2 ;PAGE IS GLOBAL PAGE TABLE
2E 11 0022 255 BRB 50$ ;PAGE IS NOT SYSTEM PAGE TYPE, COULD BE GLOBAL PAGE TABLE
0024 256 :
0024 257 : OR PROCESS HEADER PAGE
0024 258 :
0024 259 :
0024 260 NOTSYSTEM:
50 50 F7 8F 78 0024 261 ASHL #-9,RO,RO ;SCALE VA DIFFERENCE TO PAGE NUMBER
55 6C A4 D0 0029 262 MOVL PCB$L_PHD(R4),R5 ;ADDRESS OF PROCESS HEADER
50 0000'CF C6 002D 263 DIVL W$WP$GL_BSLO$SZ,RO ;PROCESS HEADER INDEX
42 A5 50 B1 0032 264 CMPW RO,PHDSW_PHVINDEX(R5) ;THIS PROCESS' HEADER?
52 08 90 0036 265 BNEQ GPGTBL ;BRANCH IF NOT, MAYBE GLOBAL PAGE TABLE
52 08 90 0038 266 MOVB #WSL$C_PPGTBL,R2 ;ASSUME PROCESS PAGE TABLE
1E 11 0038 267 BRB 70$ ;PAGE FAULT MONITORING ENABLED FOR THIS PROCESS
003D 268 :
003D 269 : PAGE FAULT MONITORING ENABLED FOR THIS PROCESS
003D 270 :
003D 271 PGFMONITOR:
FFCO' 30 003D 272 BSBW PFMSMON ;CALL THE RECORDING ROUTINE
5E 11 0040 273 BRB PGFMONITOR1 ;RETURN IN LINE
0042 274 :
0042 275 : PAGE FAULT FOR SYSTEM SPACE VIRTUAL ADDRESS
0042 276 : R2 = FAULT VA, LOW BITS CLEARED
0042 277 : R4 = PROCESS PCB ADDRESS

```

53 52 15 09 EF 0042 278 :  
50 52 0000'CF C3 0042 279 SYSTEMSPACE:  
52 D5 18 004D 280 EXTZV #VA\$V VPN,#VASS VPN,R2,R3 ;PAGE NUMBER IN SYSTEM SPACE  
54 0000'CF DE 0052 281 SUBL3 W^SWPSGL BALBASE,R2,R0 ;ABOVE BASE OF BALANCE SET SLOTS?  
55 6C A4 D0 0057 282 BGEQ NOTSYSTEM ;BRANCH IF NOT SYSTEM PAGE TYPE  
53 0000'DF43 DE 005B 283 40\$: MOVB #WSLSC SYSTEM,R2 ;SYSTEM PAGE  
57 11 0061 284 50\$: MOVAL W^MMGSAL SYSPCB,R4 ;ADDRESS OF SYSTEM PCB  
0063 285 MOVL PCB\$L PHD(R4),R5 ;ADDRESS OF SYSTEM PROCESS HEADER  
0063 286 70\$: MOVAL @W^MMGSGL SPTBASE[R3],R3 ;ADDRESS OF PAGE TABLE ENTRY  
0063 287 BRB GETPAGELOC  
0063 288  
0063 289 : THIS IS A PROCESS PAGE TABLE FAULT  
0063 290 :  
0063 291 :  
0063 292 PPGTBL:  
52 53 000001FF 8F CB 0063 293 BICL3 #^X1FF,R3,R2 ;R2 = FAULT VA  
D5 11 006B 294 BRB SYSTEMSPACE  
006D 295  
006D 296 .DSABL LSB

.SBTTL EXCEPTION ENTRY POINT - PAGE TYPE DISPATCHER

```

      006D 298 .ALIGN LONG
      006D 299 MMGSPAGEFAULT::: 
      006D 300
      0070 301 PUSHL R5 ;SAVE R5
      0070 302 PUSHL R4 ;SAVE R4
      0070 303 CMPZV #PSL$V_IPL,#PSLSS_IPL,<12+8>(SP),#IPLS_ASTDEL ;CHECK FAULT IPL
      0070 304 BGTR IPLHI ;BRANCH IF IPL IS TOO HIGH
      0070 305 PUSHL R3 ;SAVE R3
      0070 306 SETIPL #IPLS_SYNCH ;LOCK THE DATA BASE
      0070 307 MTPR #IPLS_SYNCH,S^#PRS_IPL
      007E 308 PUSHL R2 ;SAVE R2
      0081 309 MOVL W^SCH$GL_CURPCB,R4 ;R4 = ADDRESS OF PCB
      0083 310 PUSHL R1 ;SAVE R1
      0088 311 PUSHL R0 ;SAVE R0
      008A 312 BICL3 #^X1FF,FLTVA(SP),R2 ;R2 = VA OF FAULT (LOW BITS CLEAR)
      0095 313 BLSS SYSTEMSPACE ;BRANCH IF SYSTEM SPACE ADDRESS
      0097 314 MOVL PCBSL_PHD(R4),R5 ;R5 = ADDRESS OF HEADER
      009B 315 BBS #PHD$V_PFMFLG,PHD$W_FLAGS(R5),PGFMONITOR ;BRANCH IF PAGE FAULT MONIT
      00A0 316 PGFMONITOR1:
      00A0 317 EXTZV #VASV_VPN,#VASS_VPN,R2,R3 ;VIRTUAL PAGE IN P0 OR P1 SPACE
      00A5 318 BBC #VASV_P1,R2,POADDR ;BRANCH IF P0 SPACE
      00A9 319 MOVAL @PHD$C_P1BR(R5)[R3],R3 ;GET SYS VIRT ADR OF PTE FOR P1 SPACE
      00AF 320 BRB GETPAGELOC
      00B1 321 VALID: BRW PGFCOMPLETE ;IF VALID, JUST EXIT
      00B4 322 POADDR: MOVAL @PHD$L_POBR(R5)[R3],R3 ;GET SYS VIRT ADR OF PTE FOR P0 SPACE
      00BA 323 :
      00BA 324 : R2 = VA (LOW BITS = PAGTYP), R3 = SVAPTE
      00BA 325 :
      00BA 326 GETPAGELOC:
      00BA 327 EXTZV #VASV_VPN,#VASS_VPN,R3,R0 ;INDEX TO SPT ENTRY
      00BF 328 MOVAL @W^MMG$GL_SPTBASE[R0],R1 ;ADDRESS OF SPTE FOR PAGE TABLE
      00C5 329 TSTL (R1) ;IS SPTE VALID?
      00C7 330 BGEQ PPGTBL ;BRANCH IF NOT, FAULT IT
      00C9 331 BICL3 #^C<PTESM_VALID,-;CHECK VALID BIT
      00D1 332 !PTESM_TYP1 !PTESM_TYP0 -;GET PTE TYPE BITS
      00D1 333 !PTESM_PGFLVB>,(R3),R0 ;AND PFN/PAGE FILE VBN BITS TO R0
      00D1 334 BLSS VALID ;BRANCH IF VALID
      00D3 335 :
      00D3 336 : R0 = TYP1 ! TYP0 ! PGFLVB, VALID IS KNOWN TO BE OFF AT THIS POINT
      00D3 337 : R1 = SPT ENTRY ADDRESS FOR PAGE TABLE PAGE
      00D3 338 : R2 = VA (LOW BITS = PAGTYP)
      00D3 339 : R3 = SVAPTE
      00D3 340 :
      00D3 341 INCL W^PMSS$GL_FAULTS ;COUNT ALL THE PAGE FAULTS
      00D3 342 INCL PHD$L_PAGEFLTS(R5) ;COUNT PROCESS' PAGE FAULTS
      00D7 343 PUSHL R3 ;SAVE NEEDED VOLATILE REGISTERS
      00DA 344 PUSHL R2
      00DC 345 PUSHL R1
      00DE 346 BSBW MMG$FREWSLE ;FREE A WORKING SET LIST ENTRY
      00E0 347 BLBC R0,RSRCWAIT_3 ;BRANCH IF HAVE TO WAIT
      00E3 348 :
      00E6 349 : ***** ALL POINTS DISPATCHED TO FROM HERE MUST REMEMBER THAT
      00E6 350 : ***** 0(SP) = VA (LOW BITS = PAGTYP), 4(SP) = SVAPTE
      00E6 351 :
      00E6 352 MOVL 8(SP),R3 ;SVAPTE
      00EA 353 :
    
```

00EA 354 : MUST RECHECK VALIDITY OF PAGE TABLE PAGE, SINCE FREWSLE MIGHT HAVE DISCARDED IT  
 9E D5 00EA 355 :  
 3D 18 00EA 356 TSTL @SP+  
 00EC 357 BGEQ PPGTBL\_2 ;IS PAGE TABLE PAGE VALID?  
 00EE 358 :  
 00EE 359 : MUST FETCH PAGE TABLE ENTRY CONTENTS AGAIN SINCE DEAD PAGE TABLE SCAN  
 00EE 360 IN FREWSLE MIGHT HAVE DISCARDED THE PAGE. NOTE THAT VALID IS KNOWN 0.  
 00EE 361 :  
 50 63 7B800000 8F CB 00EE 362 BICL3 #^C<PTESM\_VALID ! -  
 00F6 363 PTESM\_TYP! PTE\$M\_TYP0 ;FETCH VALID BIT  
 2A 13 00F6 364 PTESM\_PGFVLVB>, (R3), R0 ;PTE TYPE BITS  
 00F8 365 BEQL DZRO\_PTE\_0 ;AND PGFLVB, GPTX, SECX, PFN  
 00F8 366 :BRANCH IF PAGE IS DEMAND ZERO  
 00F8 367 : FORM R1 = 4 BIT SIGN EXTENDED VIELD LOW BIT = TYP0, SIGN = TYP1  
 51 50 05 16 EE 00F8 368 :  
 55 12 00FD 369 EXTV #PTESV\_TYP0, #<PTESV\_TYP1+1-PTESV\_TYP0>, R0, R1  
 00FF 370 BNEQ NOTTRANSITION ;BRANCH IF NOT TRANSITION PAGE  
 00FF 371 :  
 00FF 372 : THIS IS A PAGE IN TRANSITION  
 00FF 373 : R0 = PFN, R2 = VA (LOW BITS = PAGTYP), R3 = SVAPTE  
 00FF 374 :  
 00FF 375 TRANSITION:  
 52 0000'DF40 03 00 EF 00FF 376 EXTZV #PFNSV\_LOC, #PFN\$S\_LOC, @W^PFNSAB\_STATE[R0], R2 ;PAGE LOCATION  
 00000002 0018'CF42 D6 0107 377 IF GT\_CAS\_MEASURE  
 0107 378 INCL W^PMSSAL\_TRANSFLT[R2] ;COUNT VARIOUS TRANSITION FAULTS  
 010C 379 .ENDC  
 010C 380 :  
 010C 381 CASE R2,<-  
 010C 382 PFNLIST,- ;ON THE FREE PAGE LIST  
 010C 383 PFNLIST,- ;ON THE MODIFIED PAGE LIST  
 010C 384 PFNLIST,- ;ON THE BAD PAGE LIST  
 010C 385 RELEASEPEND,- ;RELEASE PENDING  
 010C 386 READERR,- ;PAGE READ ERROR  
 010C 387 WRITEINPROG,- ;PAGE WRITE IN PROGRESS  
 010C 388 READINPROG- ;PAGE READ IN PROGRESS  
 010C 389  
 010C 390 >  
 06' 00 52 AF 010C CASEW R2, #0, S^#<>30001\$-30000\$>/2>-1  
 0110 30000\$: .SIGNED\_WORD PFNLIST-30000\$  
 04FC' 0110 .SIGNED\_WORD PFNLIST-30000\$  
 04FC' 0112 .SIGNED\_WORD PFNLIST-30000\$  
 04FC' 0114 .SIGNED\_WORD RELEASEPEND-30000\$  
 04FF' 0116 .SIGNED\_WORD READERR-30000\$  
 054B' 0118 .SIGNED\_WORD WRITEINPROG-30000\$  
 04FF' 011A .SIGNED\_WORD READINPROG-30000\$  
 0272' 011C .SIGNED\_WORD  
 011E 30001\$: 391 LOCBAD: BUG\_CHECK PGFLOCBAD, FATAL ;BAD PAGE LOCATION FIELD  
 FEFF 011E .WORD ^XFEFF .IIF IDN <FATAL>, <FATAL>, .WORD BUGS\_PGFLOCBAD!4  
 0004' 0120 0122 392 DZRO\_PTE 0:  
 0122 393 RSRCWAIT BRW 3: DZRO\_PTE  
 0409 31 0122 394 ADDL #3\*4, SP ;CLEAN OFF 3 LONG WORDS  
 5E 0C C0 0125 395 RESOURCEWAIT ;AND GO WAIT FOR A RESOURCE  
 0253 31 0128 396 BRW  
 012B 397 PPGTBL\_2:

PAGEFAULT  
V04-000

B 6  
- TRANSLATION NOT VALID EXCEPTION HANDLE 16-SEP-1984 00:43:02 VAX/VMS Macro V04-00  
EXCEPTION ENTRY POINT - PAGE TYPE DISPA 5-SEP-1984 03:45:49 [SYS.SRC]PAGEFAULT.MAR;1 Page 11  
(5)

03 FF33 BA 012B 399 POPR #^M<R0,R1>  
03 FF33 31 012D 400 BRW PPGTBL ;CLEAN OFF 2 LONG WORDS, SCRATCH R0,R1  
;FAULT A PROCESS PAGE TABLE

0130 402 .SBTTL PAGE FILE, SECTION TABLE INDEX, OR GLOBAL PAGE  
 0130 403 :  
 0130 404 : GLOBAL PAGE, MASTER PTE VALID.  
 0130 405 : R0 = MASTER PTE CONTENTS (VALID, MODIFY AND PFN BITS)  
 0130 406 : 0(SP) = PROCESS VA (LOW BITS = PAGTYP), 4(SP) = SLAVE PTE ADDRESS  
 0130 407 :  
 0130 408 GBLVALID:  
 003C'CF D6 0130 409 INCL W^PMSSGL\_GVALID ;UPDATE GLOBAL VALID COUNTER  
 52 6E 7D 0134 410 MOVQ (SP),R2 ;R2=VA (LOW BITS = PAGTYP), R3=SVPATE  
 50 50 15 00 0950 0137 411 PUSHL RO ;SAVE MASTER PTE  
 0139 412 EXTZV #PTE\$V\_PFN,#PTESS\_PFN,R0,R0 ;PAGE FRAME NUMBER  
 013E 413 BSBW MMGSMAREWSLE ;MAKE A WORKING SET LIST ENTRY  
 0E BA 0141 414 POPR #^M<R1,R2,R3> ;R1=MASTER PTE, R2=VA, R3=SVPATE  
 04FC 31 0143 415 BRW SETSLAVEPTE ;SET PROCESS' PTE AND EXIT  
 0146 416 :  
 0146 417 : DEMAND ZERO GLOBAL SECTION WITH PAGE FILE BACKING STORE  
 0146 418 :  
 50 00400000 8F 007A D0 0146 419 GBLDZRO\_PGFL:  
 31 014D 420 MOVL #PTE\$M\_TYP0,R0 ;ADD A TYPO BIT TO THE MASTER PTE (0)  
 0150 421 BRW GBLDZRO ;GO JOIN COMMON CODE  
 0150 422 :  
 0150 423 : BAD MASTER PAGE TABLE ENTRY FORMAT FOR A GLOBAL PAGE  
 0150 424 :  
 0150 425 GBLBAD: BUG\_CHECK PGFGBLBAD,FATAL ;BAD MASTER PTE FORMAT FOR GLOBAL PAGE  
 0004' FEFF 0152 .WORD ^XF<sup>E</sup>FF  
 .IIF IDN <FATAL>,<FATAL> , .WORD BUG\$\_PGFGBLBAD!4  
 0154 426 :  
 0154 427 : PAGE IS NOT A TRANSITION OR DEMAND ZERO PAGE  
 0154 428 : R0 = LOW 23 BITS OF PTE AND TYPE BITS (PAGE FILE VBN, GPTX OR STX)  
 0154 429 : R1 = RESULT OF EXTV ABOVE, CONDITION CODES SET FROM EXTV  
 0154 430 : 0(SP) = VA (LOW BITS = PAGTYP), 4(SP) = SVPATE  
 0154 431 :  
 0154 432 NOTTRANSITION:  
 00 50 7A 19 0154 433 BLSS NOTGLOBAL ;BRANCH IF TYP1 SET, NOT A GLOBAL PAGE  
 53 0000'DF40 E5 0156 434 BBCC #PTE\$V\_TYP0,R0,10\$ ;LEAVE JUST GLOBAL PAGE TABLE INDEX  
 51 63 0A 9C DE 015A 435 10\$: MOVAL @W^MMG\$GL\_GPTBASE[R0],R3 ;ADDRESS OF MASTER PTE  
 0160 436 ROTL #<32-<PTE\$V\_OWN-WSLSV\_PAGTYP>,(R3),R1 ;OWNER FIELD TO LOW BITS  
 0164 437 :  
 0164 438 : MASTER PTE OWNER FIELD CONTAINS THE VALUE PFNSC\_GLOBAL OR PFNSC\_GBLWRT  
 0164 439 :  
 50 6E 51 F1 8F 8B 0164 440 BICB3 #^C<WSL\$M\_PAGTYP>,R1,(SP) ;SET PAGE TYPE FIELD  
 63 7B800000 8F CB 0169 441 BICL3 #^C<PTE\$M\_VALID ! - ;GET THE VALID BIT  
 0171 442 PTE\$M\_TYP1 ! PTE\$M\_TYP0 ; - ;THE PTE TYPE BITS  
 0171 443 PTE\$M\_PGFLVB>,(R3),R0 ;AND PAGE FILE VBN, OR STX  
 BD 19 0171 444 BLSS GBLVACID ;BRANCH IF MASTER PTE IS VALID  
 D1 13 0173 445 BEQL GBLDZRO\_PGFL ;MASTER PTE IS DEMAND ZERO  
 51 50 05 16 EE 0175 446 EXTV #PTE\$V\_TYP0,#<PTE\$V\_TYP1+1-PTE\$V\_TYP0>,R0,R1  
 017A 447 :  
 017A 448 : R1 = 0 IF TYP1 AND TYPO ARE BOTH ZERO  
 017A 449 : R1 = NEGATIVE IF TYP1 IS SET  
 017A 450 : R1<0> = 1 IF TYPO IS SET  
 017A 451 :  
 83 13 017A 452 BEQL TRANSITION ;BRANCH IF GLOBAL TRANSITION  
 D2 14 017C 453 BGTR GBLBAD ;BRANCH IF GLOBAL AGAIN, ERROR  
 017E 454 :  
 017E 455 : MASTER PAGE TABLE ENTRY IS A SECTION OR PAGE FILE ADDRESS  
 017E 456 :

45 50 08 51 E9 017E 457 20\$: BLBC R1,30\$ ;BRANCH IF PAGE FILE  
06 50 11 EO 0181 458 BBS #PTESV\_DZRO,RO,GBLDZRO ;BRANCH IF DEMAND ZERO GLOBAL SECTION  
06 50 10 EO 0185 459 BBS #PTESV\_CRF,RO,GBLCRF ;BRANCH IF COPY ON REFERENCE  
0189 460 :  
0189 461 : GLOBAL SECTION (NOT CRF OR DZRO) OR PAGE FILE BACKING STORE ADR  
0189 462 :  
50 04 AE D0 0189 463 30\$: MOVL 4(SP),R0 ;SAVE SLAVE PTE ADR INDICATING GLOBAL  
4A 11 018D 464 BRB GBLNOTRESIDENT ;  
018F 465 :  
018F 466 : GLOBAL COPY ON REFERENCE PAGE  
018F 467 :  
018F 468 GBLCRF:  
50 6E 94 018F 469 CLRB (SP) ;SAY PAGE IS PROCESS PAGE  
53 D0 0191 470 MOVL R3,RO ;MASTER PTE ADDRESS  
43 11 0194 471 BRB GBLNOTRESIDENT ;

```

      0196 473 :SBTTL PAGE NOT RESIDENT, QUEUE A READ REQUEST
      0196 474 :ENABL LSB

      0196 475 :
      0196 476 : MUST WAIT FOR AN I/O REQUEST PACKET
      0196 477 :
      0196 478 :
      0196 479 IRPWAIT_3:
      51 02 BA 0196 480 POPR #^M<R1> :CLEAN OFF 1 LONG WORD
      03 9A 0198 481 MOVZBL #RSNS_NPDYNMEM,R1 :NON PAGED DYNAMIC MEMORY RESOURCE NUMBER
      OC BA 019B 482 10$: POPR #^M<R2,R3> :CLEAN OFF 2 LONG WORDS
      01DE 31 019D 483 BRW RESOURCEWAIT :WAIT FOR RESOURCE IN R1
      01A0 484 :
      01A0 485 : THIS PAGE READ WOULD EXCEED THIS PROCESS' DIRECT I/O QUOTA.
      01A0 486 : WAIT UNTIL SOME OF HIS OUTSTANDING I/O COMPLETES.
      01A0 487 :
      01A0 488 DIOCNTWAIT_2:
      51 01 9A 01A0 489 MOVZBL #RSNS_ASTWAIT,R1 :AST WAIT RESOURCE NUMBER
      F6 11 01A3 490 BRB 10$ :
      01A5 491 :
      01A5 492 .DSABL LSB
      01A5 493 :
      01A5 494 :
      01A5 495 : NO I/O PACKETS ON THE SIDE LIST, MUST ALLOCATE ONE FROM NON-PAGED POOL
      01A5 496 : 0(SP) = PLACE TO STORE ADDRESS OF PACKET, TOTAL OF 3 LONG WORDS ON STACK
      01A5 497 :
      01A5 498 GET_IRP:
      51 6E 50 D0 01A5 499 MOVL R0,(SP) :SAVE REGISTER
      C4 8F 9A 01A8 500 MOVZBL #IRPSC_LENGTH,R1 :SIZE OF I/O PACKET
      FE51 30 01AC 501 BSBW EXESALNONPAGED :ALLOCATE THE PACKET
      E4 50 E9 01AF 502 BLBC R0,IRPWAIT_3 :BRANCH IF NONE AVAILABLE
      01 BA 01B2 503 POPR #^M<R0> :RESTORE SAVED REGISTER
      52 DD 01B4 504 PUSHL R2 :I/O PACKET ADDRESS
      6A 11 01B6 505 BRB GOT_IRP :REJOIN THE MAIN FLOW
      01B8 506 :
      01B8 507 : MUST WAIT FOR A FREE PAGE, 5 LONG WORDS ON STACK, FIRST 2 ARE GARBAGE
      01B8 508 : 8(SP) = I/O REQUEST PACKET ADDRESS TO BE DEALLOCATED, LAST 2 ARE SCRATCH
      01B8 509 :
      01B8 510 FREPAGWAIT_5:
      03 BA 01B8 511 POPR #^M<R0,R1> :SCRATCH R0,R1
      07 BA 01BA 512 POPR #^M<R0,R1,R2> :I/O PACKET ADDRESS TO R0
      08 A0 C4 8F 9A 01BC 513 MOVZBL #IRPSC_LENGTH,IRPSW_SIZE(R0) :SET PACKET SIZE AND CLEAR TYPE
      FE3C 30 01C1 514 BSBW EXESDEANONPAGED :AND DEALLOCATE IT
      01B0 31 01C4 515 BRW FREEPAGEWAIT
      01C7 516 :
      01C7 517 : DEMAND ZERO PROCESS SECTION PAGE
      01C7 518 :
      0364 31 01C7 519 DZRO_PROC_SEC:
      01CA 520 BRW DZRO_PTE
      01CA 521 :
      01CA 522 : DEMAND ZERO GLOBAL SECTION PAGE
      01CA 523 :
      01CA 524 GBLDZRO:
      51 63 D0 01CA 525 MOVL (R3),R1 :MASTER PTE CONTENTS
      0362 31 01CD 526 BRW DZRO_GBL_SEC
      01D0 527 :
      01D0 528 : PAGE IS NOT A GLOBAL PAGE
      01D0 529 : R1<31>=TYP1, R1<0>=TYPO, R0 = TYP1 ! TYPO ! PGFLVB
  
```

04 51 E9 01D0 530 ;  
 F0 50 11 E0 01D0 531 NOTGLOBAL:  
 50 D4 01D3 532 BLBC R1,10\$ ;BRANCH IF NOT SECTION PAGE  
 01D7 533 BBS #PTE\$V\_DZRO,R0,DZRO\_PROC SEC ;BRANCH IF DEMAND ZERO PROCESS SECTION  
 01D9 534 10\$: CLRL R0 ;INDICATE NO SLAVE PAGE TABLE ENTRY  
 01D9 535 : 0(SP) = VA (LOW BITS = PAGTYP), 4(SP) = SVAPTE (SLAVE IF GLOBAL)  
 01D9 536 : R0 = MASTER PTE ADDRESS IF GLOBAL CRF  
 01D9 537 : = SLAVE PTE ADDRESS IF GLOBAL NOT CRF  
 01D9 538 : = 0 IF NOT GLOBAL  
 01D9 539 :  
 01D9 540 :  
 01 35 A5 91 01D9 541 GBLNOTRESIDENT:  
 32 15 15 01DD 542 CMPB PHDSB\_PGTBPFC(R5),#1 ;IF CLUSTERING PAGE TABLE PAGES  
 52 04 AE 15 09 EF 01DF 543 BLEQ 40\$  
 51 0000'DF42 DE 01E5 544 EXTZV #VASV VPN,#VASS VPN,4(SP),R2 ;SEE IF ADJACENT PAGE TABLES  
 71 847FFFFF 8F D3 01EB 545 MOVAL @W^MMG\$GL\_SPTBASE[R2],R1 ;NEED TO BE FAULTED, GET SPT ENTRY ADR  
 01F2 546 BITL #<PTESM VALID ! -  
 01F2 547 PTESM\_TYP1 ! PTESM\_TYP0 ! -  
 01F2 548 PTESM\_PGFVLVB>,-(R1) ;CHECK PREVIOUS SPT ENTRY  
 08 A1 847FFFFF 8F 0E 14 01F2 549 BGTR 10\$ ;BRANCH IF NOT VALID, NOT DZRO  
 D3 01F4 550 BITL #<PTESM VALID ! -  
 01FC 551 PTESM\_TYP1 ! PTESM\_TYP0 ! -  
 13 15 01FC 552 PTESM\_PGFVLVB>,8(R1) ;CHECK NEXT SPT ENTRY  
 01FE 553 BLEQ 40\$ ;BRANCH IF IT IS VALID  
 01FE 554 :  
 01FE 555 : NEXT PAGE TABLE NEEDS TO BE FAULTED  
 01FE 556 :  
 52 D6 01FE 557 INCL R2 ;SET NEXT SPT INDEX  
 02 11 0200 558 BRB 20\$  
 0202 559 :  
 0202 560 : PREVIOUS PAGE TABLE NEEDS TO BE FAULTED  
 0202 561 :  
 52 52 52 09 D7 0202 562 10\$: DECL R2 ;SET PREVIOUS SPT INDEX  
 00 52 1F 9C 0204 563 20\$: ROTL #9,R2,R2 ;TURN SPT INDEX BACK INTO  
 03 BA 0208 564 BBSS #VASV SYSTEM,R2,30\$ ;SYSTEM VIRTUAL ADDRESS  
 51 0000'CF FE31 31 020C 565 30\$: POPR #^M<R0,R1> ;CLEAN STACK SCRATCH R0,R1  
 3E A1 B5 0211 566 BRW SYSTEM\$PACE ;GO FAULT THE PAGE TABLE  
 0216 567 40\$: MOVL W^SCH\$GL\_CURPCB,R1 ;COULD HAVE SYSTEM PCB IN R4  
 0219 568 TSTW PCBSW\_DIOCNT(R1) ;ENOUGH DIRECT I/O QUOTA FOR THIS READ?  
 7E 0000'DF 85 15 0219 569 : NOTE THAT BUILDPKT WILL CHARGE THE READ  
 83 1D 021B 570 BLEQ DIOCNTWAIT 2 ;BRANCH IF NO, MUST WAIT.  
 0220 571 REMQUE @W^IOC\$GL\_IRPFL,-(SP) ;GET AN I/O PACKET FROM THE SIDE LIST  
 0220 572 BVS GET\_IRP ;BRANCH IF NEED TO GET ONE FROM THE POOL  
 0222 573 : R0 = MASTER PTE ADDRESS IF GLOBAL CRF  
 0222 574 : = SLAVE PTE ADDRESS IF GLOBAL NOT CRF  
 0222 575 : = 0 IF NOT GLOBAL  
 0222 576 :  
 0222 577 : 0(SP) = I/O REQUEST PACKET ADDRESS  
 0222 578 :  
 52 04 AE 7D 0222 579 GOT\_IRP:  
 7E D4 0226 580 MOVQ 4(SP),R2 ;R2=VA (LOW BITS = PAGTYP), R3=SVAPTE  
 50 DD 0228 581 CLRL -(SP) ;INIT CRF INDICATOR TO "NOT CRF PAGE"  
 0822 30 022A 582 PUSHL R0 ;SAVE GLOBAL, GBLCRF INDICATOR  
 50 D5 022D 583 BSBW MMGSININEPFN ;ALLOCATE AND INIT A NEW PFN  
 87 19 022F 584 TSTL R0 ;PFN ALLOCATED SUCCESSFULLY?  
 53 0000'DF40 D0 0231 585 BLSS FREPAGWAIT 5 ;BRANCH IF NOT, MUST WAIT  
 586 MOVL @W^PFNSAL\_PTE[R0],R3 ;GET MASTER PTÉ ADDRESS

10 AE 53 0000'DF40 D0 0237 587 : SAME AS SLAVE UNLESS GLOBAL  
                   B6 0237 588 : SAVE FOR LATER USE  
                   0238 589 : 2ND REFERENCE FOR PAGE I/O  
                   0240 590 :  
                   0240 591 : FORM R2 = BACKING STORE ADDRESS  
                   0240 592 : 0(SP) = MASTER PAGE TABLE ENTRY ADDRESS IF GLOBAL CRF  
                   0240 593 : = SLAVE PAGE TABLE ENTRY ADDRESS IF GLOBAL NOT CRF  
                   0240 594 : = 0 IF NOT GLOBAL PAGE  
                   0240 595 : 4(SP) = 0 INITIALIZED TO 'NOT COPY ON REFERENCE'  
                   0240 596 : SET THIS TO CORRECT BACKING STORE ADDRESS IF CRF PAGE  
                   0240 597 : 8(SP) = I/O REQUEST PACKET ADDRESS  
                   0240 598 : 12(SP) = VIRTUAL ADDRESS (LOW BITS = PAGTYP)  
                   0240 599 : 16(SP) = SVAPTE, GLOBAL IF NOT GBL CRF, PROCESS IF NOT GBL OR IF GBL CRF  
                   0240 600 :  
                   51 63 867FFFFF 8F CB 0240 601 : BICL3 #^C<PTESM\_PROT ! PTESM\_OWN>, (R3), R1 ; R1 = PROT AND OWN  
                   52 63 51 CB 0248 602 : BICL3 R1, (R3), R2 ; R2 = TYP1 ! TYP0 ! PGFLVB  
                   63 51 50 C9 024C 603 : BISL3 R0, R1, (R3) ; PTE = PROT ! OWN ! PFN = TRANSITION PTE  
                   14 52 1A E4 0250 604 : BBSC #PTESV\_TYP1, R2, 20\$ ; BRANCH IF PAGE FILE OR SECTION  
                   0254 605 :  
                   0254 606 : GLOBAL COPY ON REFERENCE PAGE  
                   0254 607 :  
                   6E 52 00800000 8F C9 0254 608 : MOVL (SP)+, R3 ; GET MASTER PAGE TABLE ENTRY ADDRESS  
                   63 DD 0257 609 : BISL3 #PFNSM\_GBLBAK, R2, (SP) ; GBL BACKING STORE ADR IN CRF INDICATOR  
                   52 63 17 00 EF 0261 610 : PUSHL (R3) ; SAVE MASTER PAGE TABLE ENTRY CONTENTS  
                   OC 11 0266 611 : EXTZV #PFNSV\_BAK, #PFNSS\_BAK, (R3), R2 ; GET ADR FROM MASTER PTE  
                   0268 612 : BRB 25\$ ; TO COPY ON REFERENCE SECTION LOGIC  
                   0268 613 :  
                   0268 614 : PAGE FILE OR SECTION ADDRESS  
                   0268 615 :  
                   18 52 16 E1 0268 616 : 20\$: BBC #PTESV\_TYPO, R2, 40\$ ; BRANCH IF PAGING FILE  
                   24 52 10 E1 026C 617 : BBC #PTESV\_CRF, R2, 50\$ ; BRANCH IF NOT CRF  
                   0270 618 :  
                   0270 619 : COPY ON REFERENCE SECTION TABLE ENTRY  
                   0270 620 :  
                   04 AE 52 D0 0270 621 : MOVL R2, 4(SP) ; SAVE BACKING ADDRESS IN CRF INDICATOR  
                   1C A5 D0 0274 622 : 25\$: MOVL PHDSL\_PAGFIL(R5), - ; NULL PAGE FILE ADDRESS  
                   0000'DF40 0277 623 : @W^PFNSAL\_BAK[R0] ; 'NOT YET ALLOCATED' PAGING FILE ADR  
                   86 8F 88 027B 624 : BISB #<PFNSM MODIFY ! PFNSC\_RDINPROG>, -  
                   0000'DF40 027E 625 : @W^PFNSAB\_STATE[R0] ; FORCE MODIFY BIT, READ IN PROGRESS  
                   1C 11 0282 626 : BRB 60\$  
                   0284 627 :  
                   0284 628 : PAGING FILE BACKING STORE ADDRESS  
                   0284 629 :  
                   51 55 D0 0284 630 : 40\$: MOVL R5, R1  
                   6E D5 0287 631 : TSTL (SP) ; IS IT GLOBAL  
                   05 13 0289 632 : BEQL 45\$ ; NO  
                   51 0000'DF DE 028B 633 : MOVAL @W^MMGSGL\_SYSPHD, R1 ; GET SYSTEM HEADER  
                   52 1C A1 C8 0290 634 : 45\$: BISL PHDSL\_PAGFIL(R1), R2 ; SET PAGE FILE INDEX  
                   0294 635 :  
                   0294 636 : SECTION TABLE BACKING STORE ADDRESS  
                   0294 637 :  
                   0000'DF40 06 88 0294 638 : 50\$: BISB #PFNSC\_RDINPROG, @W^PFNSAB\_STATE[R0] ; READ IN PROGRESS  
                   0000'DF40 52 D0 029A 639 : MOVL R2, @W^PFNSAL\_BAK[R0] ; STORE BACKING STORE ADDRESS  
                   02A0 640 :  
                   02A0 641 : R0 = PFN  
                   02A0 642 : R2 = BACKING STORE ADDRESS  
                   02A0 643 : R3 = PAGE TABLE ENTRY ADDRESS, PROCESS ADR IF NOT GLOBAL.

				02A0 644 : GLOBAL ADDRESS IF GLOBAL OR GLOBAL CRF
				02A0 645 : R4 = PROCESS PCB IF PROCESS PAGE, PROCESS PAGE TABLE, OR GLOBAL PAGE,
				02A0 646 : = SYSTEM PCB IF SYSTEM PAGE OR GLOBAL PAGE TABLE
				02A0 647 : R5 = PROCESS HEADER ADDRESS CORRESPONDING TO THE ABOVE PCB ADDRESS
				02A0 648 : 0(SP) = MASTER PTE CONTENTS IF GLOBAL CRF (>0)
				02A0 649 : = SLAVE PTE ADDRESS IF GLOBAL NOT CRF (<0)
				02A0 650 : = 0 IF NOT GLOBAL
				02A0 651 : 4(SP) = 0 IF PAGE IS NOT COPY ON REFERENCE
				02A0 652 : = BACKING STORE ADDRESS (FOR GBL CRF TOO) IF CRF PAGE
				02A0 653 : 8(SP) = I/O REQUEST PACKET ADDRESS
				02A0 654 : 12(SP) = VIRTUAL ADDRESS (LOW BITS = PAGTYP)
				02A0 655 : 16(SP) = SVAPTE, GLOBAL IF NOT GBL CRF, PROCESS IF NOT GBL OR IF GBL CRF
				02A0 656 :
				51 08 AE D0 02A0 657 60\$: MOVL 8(SP),R1 ;ADDRESS OF I/O REQUEST PACKET
				3C A1 5D D0 02A4 658 MOVL FP,FP_SAV(R1) ;SAVE A REGISTER
				5D 51 D0 02A8 659 MOVL R1,FP ;USE THIS FOR CLUSTER CONTEXT
				40 AD 54 7D 02AB 660 MOVQ R4,PCB_SAV(FP) ;SAVE PCB, PHD ADDRESSES
				6E D5 02AF 661 TSTL (SP) ;LESS THAN 0 IF GLOBAL PAGE
				OE 13 02B1 662 BEQL 70\$ ;BRANCH IF NOT GLOBAL PAGE
				07 14 02B3 663 BGTR 65\$ ;BRANCH IF GLOBAL CRF
54	00000000'EF	DE	02B5 664 MOVAL L^MMGSAL_SYSPCB,R4 ;USE SYSTEM PCB FOR GLOBAL PAGES	
			02BC 665	
55	0000'DF	DE	02BC 666 65\$: MOVAL @W^MMGSGL_SYSPHD,R5 ;WANT THE PRIORITY FROM IT	
			02C1 667	
				02C1 668 : IF THE BACKING STORE ADDRESS IN R2 IS A GLOBAL ADDRESS, THEN R3 IS THE MASTER PTE
				02C1 669 : AND R5 IS THE SYSTEM PROCESS HEADER, OTHERWISE R3 IS THE PROCESS PTE
				02C1 670 : ADDRESS AND R5 IS THE PROCESS HEADER ADDRESS.
			02C1 671	
6D	52 7D 02C1 672 70\$: MOVQ R2,PTEDAT(FP) ;SAVE PTE DATA, AND ADDRESS			
			02C4 673	
10	AD 8E 7D 02C4 674 ASSUME ASTPRM EQ AST+4			
			02C4 675 MOVQ (SP)+,AST(FP) ;STORE PARAMETERS TO IOPOST IN IRP	
			02C8 676	
				02C8 677 : FETCH THE TRANSFER PRIORITY FROM PROCESS PCB IF PROCESS PAGE,
				02C8 678 : PROCESS PAGE TABLE, OR GLOBAL CRF PAGE. USE SYSTEM PRIORITY
				02C8 679 : IF SYSTEM PAGE, GLOBAL PAGE, OR GLOBAL PAGE TABLE.
			02C8 680	
23	AD 2F A4 90 02C8 681 MOVB PCB\$B PRIB(R4),PRI(FP) ;STORE PRIORITY OF TRANSFER IN IRP			
24	AD 0000'DF40 D0 02CD 682 MOVL @W^PFNSAL_BAK[R0],BAK(FP) ;SAVE BACKING STORE ADDRESS			
			78 8F 8B 02D4 683 BICB3 #^C<PFNSM_MODIFY ! PFNSM LOC>,- ;AND STATE BYTE INFORMATION	
22	AD 0000'DF40 6E D4 02D7 684 @W^PFNSAB_STATE[R0],STATE(FP) ;FROM PFN DATA OF FIRST PAGE			
			6E D4 02DD 685 CLRL (SP) ;THROUGH WITH IRP ADDRESS, USE FOR SCRATCH	
			02DF 686 CLRL PHVREFCADR(FP) ;WILL BE PAGE TABLE FAULT CLUSTER IF PPGTBL	
48	AD D4 02DF 687 CLRL PHVREFCADR(FP) ;ADDRESS OF PROCESS HEADER REF CNT IF PPGTBL			
			02E2 688	
			02E2 689 ASSUME PFNSC_PROCESS EQ 0	
			02E2 690 ASSUME PFNSC_SYSTEM EQ 1	
			02E2 691 ASSUME PFNSC_GLOBAL EQ 2	
			02E2 692 ASSUME PFNSC_GBLWRT EQ 3	
			02E2 693 ASSUME PFNSC_PPGTBL EQ 4	
			02E2 694 ASSUME PFNSC_GPGTBL EQ 5	
04	03 01 ED 02E2 695 CMPZV #WSL\$V PAGTYP,#WSLSS_PAGTYP,- ;IF PAGE TABLE PAGE			
			04 04 AE 02E5 696 4(SP),#PFNSC_PPGTBL ;THEN SEPARATE CLUSTER FACTOR	
			1B 19 02E8 697 BLSS 90\$ ;BRANCH IF NOT PAGE TABLE	
			16 14 02EA 698 BGTR 80\$ ;BRANCH IF GLOBAL PAGE TABLE	
			02EC 699	
			02EC 700 ; MUST RECORD A PROCESS HEADER REFERENCE FOR PAGE READ OF PROCESS HEADER PAGE	

51 51 42 A5 02EC 701 :  
 51 0000'DF41 3C 02EC 702 :  
 61 B6 02F0 703 :  
 48 AD 51 02F6 704 :  
 6E 35 A5 02FC 705 :  
 03 12 0300 706 :  
 6E 01 0302 708 80\$:  
 08A7 30 0305 709 90\$:  
 23 0000'CF 00000000'8F E0 0308 710 :  
 54 40 AD 7D 0312 711 :  
 0316 712 :  
 01 AE 0B A1 90 0316 713 :  
 51 6E 9A 031B 714 :  
 0A 14 031E 715 :  
 51 01 AE 9A 0320 716 :  
 0324 717 :  
 51 34 A5 9A 0326 718 :  
 032A 719 :  
 01 51 D1 032A 720 :  
 06 15 032D 721 :  
 5E 04 C0 032F 722 :  
 008E 31 0332 723 :  
 51 01 D0 0335 724 100\$:  
 5E 08 C0 0338 725 :  
 53 8E D0 033B 726 :  
 033E 727 :  
 033E 728 110\$:  
 033E 729 :  
 033E 730 :  
 033E 731 :  
 033E 732 : R0 = VBN, R1 = PAGE COUNT, R2 = WINDOW, R3 = SVPTE, FP = IRP  
 033E 733 :  
 033E 734 : QUEUE\_PAGE\_READ:  
 0008'CF D6 033E 735 :  
 0004'CF 51 C0 0342 736 :  
 54 0000'CF D0 0347 737 :  
 55 6C A4 D0 034C 738 :  
 0108 C5 D6 0350 739 :  
 51 51 09 78 0354 740 :  
 55 5D D0 0358 741 :  
 5D 3C A5 D0 035B 742 :  
 035F 743 :  
 035F 744 :  
 035F 745 :  
 035F 746 : R0 = VBN  
 035F 747 : R1 = NUMBER OF BYTES TO READ  
 035F 748 : R2 = WINDOW ADDRESS  
 035F 749 : R3 = SVPTE (MASTER IF GLOBAL, SLAVE IF GLOBAL CRF)  
 035F 750 : R4 = PROCESS PCB ADDRESS  
 035F 751 : R5 = I/O REQUEST PACKET ADDRESS  
 035F 752 :  
 035F 753 : IRPSL\_AST(R5)  
 035F 754 : = MASTER PTE CONTENTS IF GLOBAL CRF (>0)  
 035F 755 : = SLAVE PTE ADDRESS IF GLOBAL NOT CRF (<0)  
 035F 756 : = 0 IF NOT GLOBAL  
 035F 757 : IRPSL\_ASTPRM(R5)

MOVZWL PHDSW\_PHVINDEX(R5),R1 ;PROCESS HEADER VECTOR INDEX  
 MOVAW @W^PHVSGL\_REF[CBAS[R1]],R1 ;ADDRESS OF PROCESS HEADER REF CNT  
 INCW (R1) ;COUNT ANOTHER REFERENCE  
 MOVL R1,PHVREFCADR(FP) ;SAVE ADDRESS FOR CLUSTERING CODE  
 MOVB PHDSB\_PGTBPFC(R5),(SP) ;GET PAGE TABLE CLUSTER FACTOR  
 BNEQ 90\$ ;BRANCH IF SPECIFIED  
 MOVL #1,(SP) ;INDICATE NO CLUSTERING  
 BSBW MMGSINIBLDPKT ;SET UP REGISTERS TO CALL BUILDPKT  
 BBS #EXESV\_NOCLUSTER,W^EXESGL\_FLAGS,110\$ ;BRANCH IF CLUSTERING DISABLED  
 MOVQ PCB\_SAV(FP),R4 ;RECOVER PCB, PHD ADDRESSES  
 ASSUME SEC\$B\_PFC EQ PFL\$B\_PFC  
 MOVB SEC\$B\_PFC(R1),1(SP) ;PAGE FAULT CLUSTER FROM SECTION OR PAGE FILE CONTROL BLOCK  
 MOVZBL (SP),R1 ;SEE IF PAGE TABLE CLUSTER SPECIFIED  
 BGTR 100\$ ;BRANCH IF SPECIFIED  
 MOVZBL 1(SP),R1 ;SEE IF PAGE FILE OR SECTION TABLE  
 CLUSTER WAS SPECIFIED  
 BGTR 100\$ ;BRANCH IF IT WAS  
 MOVZBL PHDSB\_DFPFC(R5),R1 ;DEFAULT FROM PROCESS HEADER  
 PROCESS IF PROCESS OR GLOBAL PAGE  
 SYSTEM IF SYSTEM PAGE  
 CMPL R1,#1 ;CLUSTER OF 1?  
 BLEQ 110\$ ;BRANCH IF YES  
 ADDL #4,SP ;CLEAN OFF CLUSTER FACTOR SCRATCH  
 BRW TRY\_TO\_CLUSTER ;GO TRY TO CLUSTER  
 MOVL #1,R1 ;ONE PAGE READ  
 ADDL #8,SP ;CLEAN OFF VIRTUAL ADDRESS  
 MOVL (SP)+,R3 ;SVPTE FROM PFNSAL\_PTE  
 033E 731 : R0 = VBN, R1 = PAGE COUNT, R2 = WINDOW, R3 = SVPTE, FP = IRP  
 033E 732 :  
 033E 733 :  
 033E 734 : QUEUE\_PAGE\_READ:  
 INCL W^PMSSGL\_PREADIO ;COUNT PAGE READ I/O REQUESTS (SYSTEM)  
 ADDL R1,W^PMSSGL\_PREADS ;AND THE NUMBER OF PAGES READ  
 MOVL W^SCH\$GL\_CURPCB,R4 ;PCB ADDRESS  
 MOVL PCBSL\_PHD(R4),R5 ;PHD ADDRESS  
 INCL PHDSL\_PGFILTIO(R5) ;COUNT PAGE READ I/O REQUESTS (PROCESS)  
 ASHL #9,R1,R1 ;FORM BYTE COUNT TO TRANSFER  
 MOVL FP,R5 ;I/O PACKET ADDRESS  
 MOVL FP\_SAV(R5),FP ;RESTORE SAVED REGISTER  
 035F 746 : R0 = VBN  
 035F 747 : R1 = NUMBER OF BYTES TO READ  
 035F 748 : R2 = WINDOW ADDRESS  
 035F 749 : R3 = SVPTE (MASTER IF GLOBAL, SLAVE IF GLOBAL CRF)  
 035F 750 : R4 = PROCESS PCB ADDRESS  
 035F 751 : R5 = I/O REQUEST PACKET ADDRESS  
 035F 752 :  
 035F 753 : IRPSL\_AST(R5)  
 035F 754 : = MASTER PTE CONTENTS IF GLOBAL CRF (>0)  
 035F 755 : = SLAVE PTE ADDRESS IF GLOBAL NOT CRF (<0)  
 035F 756 : = 0 IF NOT GLOBAL  
 035F 757 : IRPSL\_ASTPRM(R5)

035F 758 : = BACKING STORE ADDRESS IF CRF PAGE (GBLBAK SET IF GBL CRF)  
 035F 759 : = 0 IF NOT CRF PAGE  
 035F 760 : IRPSB\_PRI(R5) = DESIRED TRANSFER PRIORITY  
 035F 761 :  
 FC9E' 30 035F 762 BSBW EXESBUILDPKTR ;BUILD AND QUEUE THE I/O PACKET  
 0362 763 :  
 0362 764 : THE FOLLOWING WAITS THE PROCESS AT THE FAULTING MODE  
 0362 765 :  
 50 0000'CF 49 7E 0362 766 PROCpag:  
 0367 767 MOVAQ W\$CH\$GQ\_PFWQ,R0 ;PAGE FAULT WAIT QUEUE ADDRESS  
 0367 768 BSBB MMG\$PGFLTWAIT\_1 ;PUT PROCESS ON PAGE FAULT WAIT QUEUE  
 5E 08 BA 0369 769 PGFEXIT:  
 0368 770 POPR #^M<R0,R1,R2,R3,R4,R5> ;RESTORE REGISTERS SAVED BY PAGE FAULT  
 0368 771 ADDL #8,SP ;CLEAN OFF THE EXCEPTION PARAMETERS  
 036E 772 :  
 036E 773 : STACK NOW CONTAINS JUST THE FAULT PC, PSL PAIR  
 036E 774 :  
 036E 775 MMG\$SVPCTX:  
 54 0000'CF FC89' 07 036E 776 SVPCTX ;SAVE PROCESS CONTEXT  
 036F 777 MOVL W\$CH\$GL\_CURPCB,R4 ;GET PCB ADDRESS  
 0374 778 BRW SCHSWAITM ;JOIN COMMON WAIT CODE FOLLOWING SVPCTX  
 0377 779 :  
 0377 780 : NO FREE PAGES AVAILABLE ON THE FREE PAGE LIST, MUST WAIT  
 0377 781 :  
 50 0000'CF 1A 7E 0377 782 FREEPAGEWAIT:  
 037C 783 MOVAQ W\$CH\$GQ\_FPGWQ,R0 ;WAIT ON FREE PAGE WAIT QUEUE  
 037C 784 BRB FREEPAGEWAIT1  
 037E 785 :  
 037E 786 : WAIT FOR RESOURCE IN R1 TO BECOME AVAILABLE  
 037E 787 :  
 1C 10 037E 788 RESOURCEWAIT:  
 E7 11 0380 789 BSBB MMG\$RESRCWAIT ;SET UP TO WAIT FOR THE RESOURCES  
 0380 790 BRB PGFEXIT ;AND EXIT TO THE SCHEDULER  
 0382 791 :  
 0382 792 : FAULT FOR PAGE WHICH IS ALREADY ON THE WAY INTO MEMORY  
 0382 793 :  
 0382 794 READINPROG:  
 OC BA 0382 795 POPR #^M<R2,R3> ;R2=VA (LOW BITS = PAGTYP), R3=SVAPE  
 0000'DF40 07 93 0384 796 ASSUME PFNSC PROCESS EQ 0  
 D6 13 0384 797 BITB #PFNSM\_PAGTYP, @W\$PFNSAB\_TYPE[R0] ;PROCESS PAGE?  
 00 0000'DF40 04 E2 038C 798 BEQL PROCpag ;BRANCH IF YES  
 0393 799 BBSS #PFNSV\_COLLISION, @W\$PFNSAB\_TYPE[R0], 10\$ ;COLLISION OCCURRED  
 50 0000'CF 7E 0393 800 10\$:  
 0398 801 MOVAQ W\$CH\$GQ\_COLPGWQ,R0 ;COLLISION PAGE WAIT QUEUE  
 18 10 0398 802 FREEPAGEWAIT1:  
 CD 11 039A 803 BSBB MMG\$PGFLTWAIT\_1 ;PLACE PROCESS ON THE COLLISION QUEUE  
 039C 804 BRB PGFEXIT ;EXIT TO THE SCEDULER  
 039C 805 :  
 039C 806 : WAIT FOR RESOURCE IN R1 TO BECOME AVAILABLE  
 039C 807 : R0,R1,R2,R3 ALTERED, R4 RETURNED WITH CURRENT PCB ADDRESS  
 039C 808 :  
 039C 809 .ENABL LSB  
 039C 810 MMG\$RESRCWAIT::  
 54 0000'CF 4C A4 51 039C 811 MOVL W\$CH\$GL\_CURPCB,R4 ;R4 = CURRENT PCB ADDRESS  
 00 0000'CF 51 E6 03A1 812 MOVL R1,PCBSL\_EFWM(R4) ;SET RESOURCE NEEDED  
 03A5 813 BBSSI R1,W\$CH\$GL\_RESMASK,10\$ ;NOTE SOMEONE WAITING  
 03AB 814 10\$:

50 0000'CF 7E 03AB 815 MOVAQ W\$CH\$GQ\_MWAIT,R0 ;WAIT ON MISCELLANEOUS QUEUE  
05 11 03B0 816 BRB 20\$ ;GO WAIT THIS PROCESS  
54 0000'CF D0 03B2 817 MMG\$PGFLTWAIT\_1:  
03B2 818 MOVL W\$CH\$GL\_CURPCB,R4 ;MUST WAIT THE PROCESS PCB  
03B7 819 MMG\$PGFLTWAIT::  
03B7 820 20\$:  
08 A0 B6 03B7 821 INCW WQH\$W\_WQCNT(R0) ;COUNT THIS PROCESS WAITING  
60 64 0E 03BA 822 INSQUE (R4),TRO ;QUEUE THIS PCB  
2C A4 60 A0 B0 03BD 823 MOVW WQH\$W\_WQSTATE(R0),PCBSW\_STATE(R4) ;SET WAIT STATE IN PCB  
05 03C2 824 RSB  
03C3 825  
03C3 826 .DSABL LSB

```

03C3 828 .SBTTL FORM A CLUSTER OF PAGES TO READ
03C3 829 :
03C3 830 : R0 = VBN IN FILE OF FIRST PAGE TO READ
03C3 831 : R1 = DESIRED CLUSTER SIZE
03C3 832 : R2 = WINDOW CONTROL BLOCK ADDRESS
03C3 833 : R4 = PCB ADDRESS, PROCESS IF PROCESS OR GLOBAL PAGE, SYSTEM IF SYSTEM PAGE
03C3 834 : R5 = PHD ADDRESS, PROCESS IF PROCESS OR GLOBAL PAGE, SYSTEM IF SYSTEM PAGE
03C3 835 : FP = I/O REQUEST PACKET ADDRESS
03C3 836 : O(SP) = VIRTUAL ADDRESS (LOW BITS = PAGTYP)
03C3 837 : 4(SP) = SVAPTE FROM PFNSAL_PTE
03C3 838 :
03C3 839 :
03C3 840 .ENABL LSB
03C3 841 :
03C3 842 TRY_TO_CLUSTER:
0C AD 8E DO 03C3 843 MOVL (SP)+,VA(FP) ;SAVE VIRTUAL ADDRESS
34 AD 50 DO 03C7 844 MOVL R0,VBN(FP) ;SET VIRTUAL BLOCK NUMBER
20 AD 51 0100 8F A9 03CB 845 :
38 AD 52 DO 03D2 846 ASSUME COUNT EQ CLUSTER+1
03D6 847 BISW3 #^X0100,R1,CLUSTER(FP) ;SET COUNT AND CLUSTER
03D6 848 MOVL R2,WINDOW(FP) ;WINDOW ADDRESS
03D6 849 :
03D6 850 : PUT PTEDAT INTO FORM OF TYP1 ! TYP0 ! PGFLVB
03D6 851 :
03D6 852 ASSUME PFNSV_PGFLX GE 24
03D6 853 ASSUME PTESV-TYP1 GE 24
03 AD 04 90 03D6 854 MOVB #PTESM_TYP1@-24,PTEDAT+3(FP) ;TURN TYP1 BACK ON, CLEAR PAGE FILE IND
53 10 AD D0 03DA 855 MOVL AST(FP),R3 ;PROCESS PTE ADR IF GBL NOT CRF
18 13 03DE 856 BEQL 30$ ;BRANCH IF NOT GLOBAL PAGE
0C 19 03E0 857 BLSS 10$ ;BRANCH IF GBL NOT CRF
03E2 858 :
03E2 859 : GLOBAL COPY ON REFERENCE PAGE
03E2 860 :
52 53 14 6E DO 03E2 861 MOVL (SP),R3 ;PROCESS PTE ADR WHEN GBL CRF
06 11 03E5 862 MOVL ASTPRM(FP),R2 ;GPTX PTE CONTENTS FOR THIS CASE
03EB 863 BRB 20$ :
00EB 31 03EB 865 CLU-END1:
03EE 866 BRW CLU-END
03EE 867 :
03EE 868 :
03EE 869 : GLOBAL PAGE NOT COPY ON REFERENCE
03EE 870 :
52 FB800000 63 DO 03EE 871 10$: MOVL (R3),R2 ;GPTX FROM PROCESS PTE
CA 872 20$: BICL #^C<PTESM_TYP1 ! PTESM_TYP0 !- ;ISOLATE PAGE TYPE
03F1 873 PTESM_GPTX>,R2 ;AND GPTX BITS
03F8 874 :
03F8 875 ASSUME GPTX_PTE EQ GPTX+4
18 AD 52 7D 03F8 876 30$: MOVQ R2,GPTX(FP) ;SET GPTX AND GPTX_PTE
28 AD 01 DO 03FC 877 MOVL #1,INC1(FP) ;INIT TO SCAN FORWARDS
0400 878 :
0400 879 .DSABL LSB
0400 880 :
0400 881 CLU_INI_INC:
30 AD 28 AD 02 78 0400 882 ASHL #2,INC1(FP),INC4(FP) ;+ OR - 4
28 AD 09 78 0406 883 ASHL #9,INC1(FP),INC512(FP) ;+ OR - 512
040C 884 CLU_NXT:

```

53 04 AD 2C AD C1 040C 885 ADDL3 INC4(FP) SVAPE(FP),R3 ;NEXT PTE TO CHECK  
 04 AD 53 D0 0412 886 MOVL R3 SVAPE(FP) ;UPDATE CONTEXT  
 04 6D 16 E0 0416 887 BBS #PTE\$V TYP0 PTEDAT(FP),20\$ ;BRANCH IF SECTION ADDRESS  
 6D 28 AD C0 041A 888 ADDL INC1(FP), PTEDAT(FP) ;INCREMENT PAGE FILE ADDRESS  
 51 53 15 09 EF 041E 889 20\$: EXTZV #VASV VPN,#VASS VPN,R3,R1 ;CHECK THAT PTE IS RESIDENT  
 0000'DF41 D5 0423 890 TSTL @W^MMGSGL\_SPTBASE[R1] ;BY MAKING SURE ITS SPTE IS VALID  
 C1 18 0428 891 BGEQ CLU END1 ;BRANCH IF IT ISN'T  
 50 63 7B800000 8F CB 042A 892 BICL3 #^CZPTESM VALID !- ;GET VALID BIT  
 0432 893 PTESM TYPT ! PTE\$M TYP0 ;PAGE TYPE BITS  
 6D 50 D1 0432 894 CMPL RO,PTEDAT(FP) ;AND PAGE FILE/GPTX BITS FROM PTE  
 B4 12 0435 895 BNEQ CLU END1 ;MUST AGREE IF THIS PAGE IS IN THE CLUSTER  
 1C AD D5 0437 896 TSTL GPTX\_PTE(FP) ;BRANCH IF AT END OF CLUSTER  
 29 13 043A 898 BEQL 60\$ ;WAS THAT THE MASTER PTE FOR A GLOBAL?  
 043C 899 ;BRANCH IF NO, IT WAS PROCESS PTE  
 043C 900 : MUST TEST THAT PROCESS PTE POINTS AT THE GPTX  
 043C 901 :  
 53 1C AD 2C AD C1 043C 902 ADDL3 INC4(FP),GPTX\_PTE(FP),R3 ;NEXT PROCESS PTE ADR  
 51 53 15 09 EF 0442 903 EXTZV #VASV VPN,#VASS VPN,R3,R1 ;CHECK THAT THIS PTE IS ACCESSIBLE  
 0000'DF41 D5 0447 904 TSTL @W^MMGSGL\_SPTBASE[R1] ;BY MAKING SURE ITS SPTE IS VALID  
 9D 18 044C 905 BGEQ CLU END1 ;BRANCH IF IT ISN'T  
 50 63 7B800000 8F CB 044E 906 BICL3 #^CZPTESM VALID !- ;GET VALID BIT  
 0456 907 PTESM TYPT ! PTE\$M TYP0 ;PAGE TYPE BITS  
 52 18 AD 28 AD C1 0456 908 CMPL INC1(FP),GPTX(FP),R2 ;AND PGFLVB/GPTX FROM PTE  
 52 50 D1 045C 909 R0,R2 ;NEXT GLOBAL PAGE TABLE INDEX  
 8A 12 045F 910 BNEQ CLU-END1 ;IN THE CLUSTER?  
 0461 911 : BRANCH IF NOT  
 18 AD 52 7D 0461 913 ASSUME GPTX\_PTE EQ GPTX+4  
 0461 914 MOVQ R2,GPTX(FP) ;UPDATE GPTX  
 0465 915 :  
 0465 916 : R1 = SPT INDEX FOR PAGE TABLE PAGE  
 0465 917 : R3 = PROCESS PTE ADDRESS  
 0465 918 :  
 0A BB 0465 919 60\$: PUSHR #^M<R1,R3> ;SAVE PROCESS PTE ADR, SPT INDEX  
 0415 30 0467 920 BSBW MMGSFR\$EWSLE ;GET A FREE WORKING SET LIST ENTRY  
 52 8E 7D 046A 921 MOVQ (SP)+,R2 ;RESTORE PROCESS PTE ADR, SPT INDEX  
 67 50 E9 046D 922 BLBC R0,CLU-END\_RESRC1 ;IF CANNOT GET ONE, END THE CLUSTER  
 0470 923 :  
 0470 924 : MUST CHECK THE SPT ENTRY FOR PROCESS PAGE TABLE IS STILL VALID  
 0470 925 : FREWSLE MIGHT HAVE DISCARDED IT FROM THE WORKING SET.  
 0470 926 :  
 0000'DF42 D5 0470 927 TSTL @W^MMGSGL\_SPTBASE[R2] ;IS SPT ENTRY FOR PT STILL VALID  
 7C 18 0475 928 BGEQ CLU END RESRC ;BRANCH IF NOT  
 52 OC AD 30 AD C1 0477 929 ADDL3 INC512(FP),VA(FP),R2 ;NEXT VIRTUAL ADDRESS  
 OC AD 52 D0 047D 930 MOVL R2,VA(FP) ;UPDATE THE CONTEXT  
 05CB 30 0481 931 BSBW MMGSININEWPFN ;ALLOC AND INIT A PFN  
 50 D5 0484 932 TSTL R0 ;IF NO PFN'S AVAILABLE  
 6B 19 0486 933 BLSS CLU END RESRC ;THEN END THE CLUSTER  
 51 21 AD 96 0488 934 INCBL COUNT(FP) ;COUNT ANOTHER PAGE IN THE CLUSTER  
 51 48 AD D0 048B 935 MOVL PHVREFCADR(FP),R1 ;PROCESS HEADER REF CNT ADR  
 048F 936 BEQL 70\$ ;IF THIS IS A PROCESS PAGE TABLE PAGE  
 02 13 048F 937 INCW (R1) ;BRANCH IF NOT A PROCESS PAGE TABLE PAGE  
 61 B6 0491 938 MOVL @W^PFNSAL\_PTE[R0],R3 ;COUNT ANOTHER PROCESS HEADER REFERENCE  
 53 0000'DF40 D0 0493 939 70\$: INCW @W^PFNSAW\_REFCNT[R0] ;PROCESS PTE ADR IF NOT GLOBAL OR IF GBL CRF  
 0499 940 : GLOBAL PTE ADR IF GLOBAL NOT CRF  
 0000'DF40 B6 0499 941 INCW @W^PFNSAW\_REFCNT[R0] ;SECOND REFERENCE FOR I/O IN PROGRESS

51 63 867FFFFF 8F CB 049E 942 BICL3 #^C<PTESM PROT ! PTESM\_OWN>, (R3), R1 :PROTECTION AND OWNER FIELDS  
 63 51 50 C9 04A6 943 BISL3 R0, R1, (R3) ;FORM TRANSITION PTE FORMAT  
 52 24 AD D0 04AA 944 MOVL BAK(FP), R2 ;BACKING STORE FROM PREV PFN  
 0E 52 16 E0 04AE 945 BBS #PTESV\_TYPO, R2, 80\$ ;BRANCH IF SECTION ADDRESS  
 51 52 09 78 04B2 946 ASHL #32-PFNSS\_BAK, R2, R1 ;IF NOT A NULL PAGE FILE ADDRESS  
 08 13 04B6 947 BEQL 80\$  
 52 28 AD CO 04B8 948 ADDL INC1(FP), R2 ;THEN INCREMENT THE ADDRESS  
 24 AD 52 D0 04BC 949 MOVL R2, BAK(FP) ;AND UPDATE THE CONTEXT  
 0000'DF40 52 D0 04C0 950 80\$: MOVL R2, @W^PFNSAL\_BAK[R0] ;SET BACKING STORE ADR FOR THIS PFN  
 0000'DF40 22 AD 88 04C6 951 BISB STATE(FP), @W^PFNSAB STATE[R0] ;USE STATE FROM PREV PFN  
 20 AD 21 AD 91 04CD 952 CMPB COUNT(FP), CLUSTER(FP) ;IS CLUSTER FULL?  
 1F 18 04D2 953 BGEQ CLU-END-RESRC ;BRANCH IF YES, QUEUE THE READ  
 FF35 31 04D4 954 BRW CLU-NXT ;NO, TRY FOR ANOTHER PAGE  
 04D7 955 CLU-END-RESRC1:  
 1A 11 04D7 956 BRB CLU-END-RESRC  
 04D9 957 :  
 04D9 958 : END OF CLUSTER  
 04D9 959 :  
 04D9 960 CLU-END:  
 01 21 AD 91 04D9 961 CMPB COUNT(FP), #1 ;IF AT LEAST 2 PAGES IN CLUSTER  
 14 14 04DD 962 BGTR CLU-END-RESRC ;THEN READ THE CLUSTER  
 28 AD 28 AD CE 04DF 963 MNEGL INCT(FP), INC1(FP) ;OTHERWISE TRY TO SCAN BACKWARDS  
 0D 14 04E4 964 BGTR CLU-END-RESRC ;UNLESS ALREADY TRIED THAT  
 04 AD 04 C2 04E6 965 SUBL #4, SVAPTE(FP) ;BACK TO STARTING SVAPTE  
 02 6D 16 E0 04EA 966 BBS #PTESV\_TYPO, PTEDAT(FP), 20\$ ;BRANCH IF SECTION PAGE  
 6D D7 04EE 967 DECL PTEDAT(FP) ;BACK TO ORIG PAGE FILE VBN  
 FF0D 31 04F0 968 20\$: BRW CLU-INI-INC  
 04F3 969 :  
 04F3 970 : SET UP TO DO THE PAGE READ  
 04F3 971 :  
 04F3 972 CLU-END-RESRC:  
 50 53 8E D0 04F3 973 MOVL (SP)+, R3 ;GET PTE ADR OF FIRST PFN IN CLUSTER  
 50 34 AD D0 04F6 974 MOVL VBN(FP), R0 ;AND ITS ASSOCIATED VBN IN THE FILE  
 51 21 AD 9A 04FA 975 MOVZBL COUNT(FP), R1 ;NUMBER OF PAGES IN THE CLUSTER  
 28 AD D5 04FE 976 TSTL INC1(FP) ;IF CLUSTER WENT BACKWARDS  
 1E 14 0501 977 BGTR 20\$ ;-(COUNT-1)  
 52 01 51 C3 0503 978 SUBL3 R1, #1, R2 ;ADJUST FILE VBN  
 50 52 C0 0507 979 ADDL R2, R0 ;AND PTE ADR  
 53 6342 DE 050A 980 MOVAL (R3)[R2], R3 ;PROCESS PTE ADDRESS FOR GLOBAL PAGE?  
 54 10 AD D0 050E 981 MOVL AST(FP), R4 ;BRANCH IF NOT GLOBAL  
 0D 13 0512 982 BEQL 20\$ ;BRANCH IF GLOBAL BUT NOT CRF  
 06 19 0514 983 BLSS 10\$ ;ADJUST GLOBAL BACKING STORE ADDRESS  
 14 AD 52 C0 0516 984 ADDL R2, ASTPRM(FP)  
 05 11 051A 985 BRB 20\$ ;ADJUST PROCESS PTE ADDRESS FOR GLOBAL PAGE  
 10 AD 6442 DE 051C 986 10\$: MOVAL (R4)[R2], AST(FP) ;GET WINDOW ADDRESS  
 52 38 AD D0 0521 987 20\$: MOVL WINDOW(FP), R2 ;GO QUEUE THE PAGE READ  
 FE16 31 0525 988 BRW QUEUE\_PAGE\_READ

0528 990 .SBTTL DEMAND ZERO PAGE  
 0528 991 :  
 0528 992 : MUST WAIT FOR FREE PAGES TO BECOME AVAILABLE  
 0528 993 :  
 0528 994 DZROFPGWAIT 5:  
 5E 14 C0 0528 995 ADDC #<5\*4>,SP ;CLEAN OFF 5 LONG WORDS  
 FE49 31 0528 996 BRW FREEPAGEWAIT ;AND GO WAIT FOR A FREE PAGE  
 052E 997 :  
 052E 998 : THIS IS A DEMAND ZERO FORMAT PAGE TABLE ENTRY, R0 = 0  
 052E 999 :  
 53 D4 052E 1000 DZRO\_PTE:  
 51 D4 052E 1001 CLRL R3 ;NO GLOBAL MASTER PTE ADDRESS  
 0530 1002 CLRL R1 ;NO MASTER PTE CONTENTS  
 0532 1003 :  
 0532 1004 : R0 = BACKING STORE ADDRESS, TYP1 ! TYPO ! PGFLVB  
 0532 1005 : R1 = MASTER PAGE TABLE ENTRY CONTENTS IF GLOBAL, 0 IF NOT  
 0532 1006 : R3 = GLOBAL PAGE TABLE ENTRY ADDRESS IF GLOBAL, 0 IF NOT  
 0532 1007 : 0(SP) = FAULT VA (LOW BITS = PAGTYP)  
 0532 1008 : 4(SP) = CORRESPONDING SVAPTE  
 0532 1009 :  
 08 BB 0532 1010 DZRO\_GBL\_SEC:  
 0534 1011 PUSHR #^M<R0,R1,R3> ;SAVE GBL PTE ADR, GBL PTE CONTENTS,  
 52 OC AE 7D 0534 1012 ;AND BACKING STORE ADDRESS  
 0514 30 0538 1013 MOVQ 12(SP),R2 ;R2=VA, R3=SVAPTE  
 50 D5 0538 1014 BSBW MMGSININEWPFN ;ALLOCATE AND INIT A PFN  
 E9 19 053D 1015 TSTL R0 ;SEE IF A PFN WAS ALLOCATED  
 0000'DF40 07 88 053F 1016 BLSS DZROFPGWAIT 5 ;BRANCH IF HAVE TO WAIT  
 52 1C A5 D0 0545 1017 BISB #PFNSC\_ACTIVE, @W^PFNSAB\_STATE[R0] ;MARK PAGE ACTIVE  
 51 8E D0 0549 1018 MOVL PHDSL\_PAGFIL(R5),R2 ;ASSUME NULL PAGE FILE BACKING STORE  
 1F 13 054C 1019 MOVL (SP)+,R1 ;GET BACKING STORE ADDRESS  
 1B 51 10 E0 054E 1020 BEQL 20\$ ;BRANCH IF DEMAND ZERO FORMAT  
 OF 51 1A E0 0552 1022 BBS #PTESV\_CRF,R1,20\$ ;BRANCH IF DZRO, CRF SECTION  
 OB 51 16 E5 0556 1023 BBS #PTESV-TYP1,R1,10\$ ;CHECK FOR GLOBAL WITH PAGE FILE BACING STOR  
 52 0000'DF DE 055A 1024 : PAGE FILE BACKING STORE GLOBAL SECTION  
 52 1C A2 D0 055F 1025 MOVAL @W^MMGSGL\_SYSPHD,R2 ;GET SYSTEM HEADER  
 08 11 0563 1026 MOVL PHDSL\_PAGFIL(R2),R2 ;BACKING STORE ADDRESS  
 52 51 FFB20000 SF CB 0565 1027 BRB 20\$  
 0000'DF40 52 D0 056D 1028 10\$: BICL3 #<PTESM\_DZRO ! ^C<PFNSM\_BAK>>,R1,R2 ;BACKING STORE ADR  
 056D 1029 :WITH DZRO SHUT OFF  
 056D 1030 20\$: MOVL R2,@W^PFNSAL\_BAK[R0] ;STORE THE BACKING STORE ADDRESS  
 0573 1031 :  
 0573 1032 : 0(SP) = MASTER PTE CONTENTS IF GLOBAL, 0 IF NOT  
 0573 1033 : 4(SP) = MASTER PTE ADDRESS IF GLOBAL, 0 IF NOT  
 0573 1034 : 8(SP) = VIRTUAL ADDRESS (LOW BITS = PAGE TYPE)  
 0573 1035 : 12(SP) = SYSTEM VIRTUAL ADDRESS OF PROCESS PAGE TABLE ENTRY  
 0573 1036 :  
 63 52 08 AE 7D 0573 1037 MOVQ 8(SP),R2 ;R2=VA (LOW BITS = PAGTYP), R3=SVAPTE  
 867FFFFF 8F CB 0577 1038 BICL3 #^C<PTESM\_PROT ! PTESM\_OWN>, -  
 7E 057E 1039 (R3),-(SP) ;PROTECTION AND OWNER FROM PTE  
 63 50 94000000 8F C9 057F 1040 BISL3 #<PTESM\_VALID ! PTESC\_KW ! -  
 0587 1041 PTESM MODIFY>,R0,(R3) ;MAKE PAGE KERNEL WRITE FOR ZEROING  
 7E 53 55 C3 0587 1042 SUBL3 R5,R3,-(SP) ;SAVE BYTE INDEX TO PTE  
 03 01 ED 0588 1043 CMPZV #WSLSV\_PAGTYP,#WSLSS\_PAGTYP,- ;IF THIS IS NOT A PROCESS PAGE  
 00 52 058E 1044 R2,#PFNSC\_PROCESS ;THEN DON'T LOWER IPL TO ZERO PAGE  
 03 12 0590 1045 BNEQ 22\$ ;STAY AT HIGHER IPL IF NOT PROCESS PAGE  
 0592 1046

12 02 DA 0592 1047 SETIPL #IPL\$\_ASTDEL ;SWAPPABLE WHILE ZEROING THE PAGE  
   0592 1048 MTPR #IPL\$\_ASTDEL,S^#PRS\_IPL  
   0595 1049 22\$: IF GT\_CAS\_MEASURE  
 00000002 0595 1050 INCL W^PMSSGL\_DZROFLTS ;COUNT DEMAND ZERO PAGE FAULTS  
   0595 1051 .ENDC  
   0599 1052  
   0599 1053  
   0599 1054 :  
   0599 1055 : \*\*\*\*\* BE AWARE THAT THE FOLLOWING CLRB ASSUMES THAT BIT 8 IS NOT  
   0599 1056 : \*\*\*\*\* IN USE FOR ANY OF THE PAGE TYPE FLAGS, ETC.  
   0599 1057 :  
 62 0200 8F 00 64 52 94 0599 1058 CLRB R2 ;CLEAR OUT THE PAGE TYPE  
   54 00 2C 059B 1059 MOVC5 #0,(R4),#0,#^X200,(R2) ;ZERO THE PAGE, PCB ADDRESS TO R1  
   51 D0 05A3 1060 MOVL R1,R4 ;RECOVER PCB ADDRESS  
   05A6 1061 SETIPL #IPL\$\_SYNCH ;NOT SWAPPABLE WHILE COMPLETING THE FAULT  
   05A6 1062 ADDL3 PCB\$L PHD(R4),(SP)+,R3 ;RE-BIAS BYTE INDEX TO PTE TO GET SVAPTE  
 50 63 8E 12 6C 12 08 DA C1 05A9 1063 BICL3 #^C<PTESM\_VALID ! PTESM MODIFY ! -  
   7BE00000 8F CB 05AE 1064 PTESM PFNS,(R3),R0 ;EVEN PFN MIGHT HAVE CHANGED  
   05B6 1065 BISL3 R0,(SP)+,(R3) ;SET PTE WITH CORRECT PROT AND OWNER  
   05BA 1066 INVALID 8(SP),R2 ;INVALIDATE TRANSLATION BUFFER  
   05BA 1067 MOVL 8(SP),R2  
   05B8 1068 MTPR R2,S^#PRS\_TBIS ;SEE IF DZRO GLOBAL  
   05C1 1069 05C4 1068 BEQL 60\$ ;BRANCH IF NOT, BOTH ARE ZERO  
 62 51 13 05C6 1069 CMPL R1,(R2) ;STILL THE SAME PTE CONTENT?  
   2D 13 05C9 1070 BEQL 40\$ ;BRANCH IF YES, NO RACE TO ZERO THE PAGE  
   05CB 1071 :  
   05CB 1072 : MORE THAN ONE PROCESS STARTED TO ZERO THIS GLOBAL DEMAND ZERO PAGE  
   05CB 1073 : THIS PROCESS LOST THE RACE, CLEAN UP AND REFault.  
   05CB 1074 :  
 50 50 15 00 EF 05CB 1075 EXTZV #PTESV\_PFN,#PTESS\_PFN,R0,R0 ;ISOLATE THE PFN  
   6E 50 D0 05D0 1076 MOVL R0,(SP) ;AND SAVE IT  
   05D3 1077 MOVL PCB\$L PHD(R4),R5 ;GET PROCESS HEADER ADDRESS  
   0000'DF40 B6 05D7 1078 INCW @W^PFNSAW REFCNT[R0] ;RELEASE WSLE AND SHRCNT, BUT NOT PFN  
   51 10 A5 3C 05DC 1079 MOVZWL PHDSW WSNEXT(R5),R1 ;WORKING SET LIST INDEX  
   52 08 AE D0 05E0 1080 MOVL 8(SP),R2 ;VIRTUAL ADDRESS (LOW BITS = PAGTYP)  
   032E 30 05E4 1081 BSBW MMG\$FREWSLX ;FREE THE WORKING SET LIST ENTRY  
   04 50 E8 05E7 1082 BLBS R0,30\$ ;BRANCH IF SUCCESSFUL  
   05EA 1083 :  
   05EA 1084 : FREWSLX COULD ONLY RETURN FAILURES STATUS IF PAGE FILE BACKING STORE  
   05EA 1085 : NEEDED TO BE RESERVED AND THERE WAS NONE AVAILABLE. THIS PAGE ALREADY  
   05EA 1086 : HAS BACKING STORE, SO THIS CANNOT HAPPEN.  
   05EA 1087 :  
   05EA 1088 BUG\_CHECK FREWSLX,FATAL  
   FF FEFF WORD ^XFEFF  
   0004' 05EC .IIF IDN <FATAL>,<FATAL> , .WORD BUGS\_FREWSLX!4  
   05EE 1089 :  
 5E F40D' 09 BA 05EE 1090 30\$: POPR #^M<R0,R3> ;GET PFN AND MASTER PTE ADDRESS  
   08 30 05F0 1091 BSBW MMG\$RLPFNSAVPTE ;RELEASE PFN, SAVE PTE CONTENTS  
   OE 11 05F3 1092 ADDL #8,SP ;CLEAN OFF VA, PROCESS PTE  
   05F6 1093 BRB PGFCOMPLETE  
   05F8 1094 :  
   05F8 1095 : GLOBAL DEMAND ZERO PAGE, MAKE MASTER PTE VALID TOO  
   05F8 1096 :  
 51 867FFFFF 8F CA 05F8 1097 40\$: BICL #^C<PTESM\_PROT ! PTESM\_OWN>,R1 ;MASTER PTE PROTECTION AND OWNER

62 51 50 C9 05FF 1098      BISL3 R0,R1,(R2)      ;SET MASTER PTE VALID  
5E 10 C0 0603 1099 60\$: ADDL #4\*4,\$P      ;CLEAN OFF 4 LONG WORDS  
0606 1100 .dsabl lsb  
0606 1101 PGFCOMPLETE:  
5E 3F BA 0606 1102 POPR #^M<R0,R1,R2,R3,R4,R5> ;RESTORE THE REGISTERS  
08 C0 0608 1103 ADDL #8,SP      ;CLEAN OFF THE EXCEPTION PARAMETERS  
02 060B 1104 REI      ;AND RETURN FROM THE EXCEPTION

```

060C 1106 .SBTTL FREE, MODIFIED, OR BAD PAGE LIST, RELEASE PENDING
060C 1107 :
060C 1108 : THIS IS A FAULT OFF THE FREE, MODIFIED, OR BAD PAGE LISTS
060C 1109 : R0 = PFN, R2 = LISTID,
060C 1110 : 0(SP) = VA (LOW BITS = PAGTYP), 4(SP) = SVAPTE
060C 1111 :
060C 1112 .ENABL LSB
060C 1113 PFNLIST:
F9F1' 30 060C 1114 BSBW MMGSREMPFN ;REMOVE PFN FROM LIST
060F 1115 WRITEINPROG:
060F 1116 RELEASEPEND:
52 6E 7D 060F 1117 MOVQ (SP),R2 ;R2 = VA, R3 = SVAPTE
047C 30 0612 1118 BSBW MMGSMAKEWSLE ;MAKE A WORKING SET LIST ENTRY
52 8E 7D 0615 1119 MOVQ (SP)+,R2 ;R2=VA, R3=SVAPTE
0618 1120 :
0618 1121 : SET PAGE ACTIVE AND VALID
0618 1122 :
51 0000'DF40 D0 0618 1123 MOVL @W^PFNSAL_PTE[R0],R1 ;GET MASTER PTE ADDRESS
061E 1124 :
061E 1125 ASSUME PFNSV_DELCON EQ PFNSV_LOC+PFN$S_LOC+1 ;DELCON IS 2ND BIT TO LEFT OF
05 00 07 F0 061E 1126 INSV #PFNSC_ACTIVE,#PFNSV_LOC,#PFN$S_LOC+2,- ;BIT IN-BETWEEN IS FOR LOC EXPANSION
0000'DF40 0622 1127 #PFNSC_ACTIVE,#PFNSV_LOC,#PFN$S_LOC+2,- ;SET PAGE ACTIVE
00 61 1F E2 0626 1128 BBSS #PTESV_VA[ID,(R1),50$] ;AND CLEAR DELCON
04 52 91 062A 1129 CMPB R2,#WS[SC_GLOBAL] ;SET VALID BIT
06 18 062D 1130 50$: BGEQ 100$ ;GLOBAL OR PAGE TABLE PAGE?
3F BA 062F 1131 POPR #^M<R0,R1,R2,R3,R4,R5> ;BRANCH IF YES
5E 08 C0 0631 1132 ADDL #8,SP ;RESTORE SAVED REGISTERS
02 0634 1133 REI ;CLEAN OFF THE EXCEPTION PARAMETERS
0635 1134 :
0635 1135 : AND RETURN FROM FAULT
0635 1136 : GLOBAL PAGE OR PAGE TABLE PAGE
0635 1137 :
08 52 91 0635 1138 100$: CMPB R2,#WSLSC_PPGTBL ;PROCESS PAGE TABLE PAGE?
16 18 0638 1139 BGEQ 120$ ;BRANCH IF PROCESS OR GLOBAL PAGE TABLE
063A 1140 :
063A 1141 : GLOBAL PAGE
063A 1142 :
51 61 7BE00000 8F CB 063A 1143 BICL3 #^C<PTESM_VALID ! PTESM_MODIFY ! PTESM_PFN>,(R1),R1 ;MASTER PTE
0642 1144 :
0642 1145 : R1 = VALID AND PFN BITS TO STORE INTO SLAVE PTE
0642 1146 : R3 = SLAVE PTE ADDRESS
0642 1147 :
52 63 867FFFFF 8F CB 0642 1148 SETSLAVEPTE:
63 52 51 C9 0642 1149 BICL3 #^C<PTESM_PROT ! PTESM_OWN>,(R3),R2 ;PROTECTION AND OWNER FROM SLAVE
DF 11 064A 1150 BISL3 R1,R2,(R3) ;STORE THE NEW SLAVE PTE
064E 1151 BRB 60$ ;AND EXIT THROUGH COMMON CODE
0650 1152 :
0650 1153 : PROCESS OR GLOBAL PAGE TABLE
0650 1154 :
DD 14 0650 1155 120$: BGTR 60$ ;BRANCH IF GLOBAL PAGE TABLE
51 DD 0652 1156 PUSHL R1 ;SAVE REGISTER AROUND THE CALL
0549 30 0654 1157 BSBW MMGSDECPHDREF ;ONE LESS LOCK ON HDR SPTE
02 BA 0657 1158 POPR #^M<R1> ;RESTOR REGISTER
D4 11 0659 1159 BRB 60$ ;AND EXIT THROUGH COMMON CODE
065B 1160 :
065B 1161 .DSABL LSB
065B 1162 :

```

065B 1163 : FAILED TO READ THE DESIRED PAGE, RELEASE THE PFN AND ISSUE AN EXCEPTION  
 065B 1164 : R0 = PAGE FRAME NUMBER  
 065B 1165 : R4 = PROCESS CONTROL BLOCK ADDRESS  
 065B 1166 : R5 = SYSTEM ADDRESS OF PROCESS HEADER  
 065B 1167 : 0(SP) = VIRTUAL ADDRESS (LOW BITS = PAGE TYPE)  
 065B 1168 : 4(SP) = SYSTEM VIRTUAL ADDRESS OF PAGE TABLE ENTRY  
 065B 1169 :  
 065B 1170 READERR:  
 OC BA 065B 1171 POPR #^M<R2,R3> ;R2=VA (LOW BITS = PAGTYP), R3=SVPATE  
 065D 1172 PFN REFERENCE -  
 065D 1173 MOVZWL <@W^PFNSAx\_WSLX[R0],R1>,- ;GET WORKING SET LIST INDEX IF NOT G  
 065D 1174 LONG OPCODE=MOVL -  
 065D 1175 IMAGE=SYS NONPAGED  
 065D .SAVE LOCAL\_BLOCK  
 065D .PSECT SABSS\$-ABS  
 00000000 004C .=0  
 0000065D .RESTORE  
 065D .SAVE PSECT LOCAL\_BLOCK  
 00000000 0000 0000065D' 0000 .PSECT Z\$INIT\$PFN\_FIXUP\_TABLE  
 3C 0004 .ADDRESS ...PFN  
 DO 0005 .BYTE OPS MOVZWL  
 0000065D .BYTE OPS MOVL  
 51 0000'DF40 3C 065D .RESTORE PSECT  
 065D 1176 MOVZWL @W^PFNSAx\_WSLX[R0],R1  
 0663 1177 ASSUME PFNSC\_PROCESS EQ 0  
 0663 1178 ASSUME PFNSC\_SYSTEM EQ 1  
 0663 1179 ASSUME PFNSC\_GLOBAL EQ 2  
 0663 1180 ASSUME PFNSC\_GBLWRT EQ 3  
 0663 1181 ASSUME PFNSC\_PPGTBL EQ 4  
 0663 1182 ASSUME PFNSC\_GPGTBL EQ 5  
 02 52 03 01 EC 0663 1183 CMPV #WSL\$V\_PAGTYP,#WSLSS\_PAGTYP,R2,#PFNSC\_GLOBAL ;IS PAGE GLOBAL?  
 18 19 0668 1184 BLSS 20\$: ;BRANCH IF NOT  
 066A 1185 :  
 066A 1186 : GLOBAL PAGE DOES NOT HAVE WORKING SET LIST INDEX IN WSLX ARRAY  
 066A 1187 : MUST SCAN THE PROCESS' WORKING SET LIST FOR THE VIRTUAL ADDRESS  
 066A 1188 :  
 55 00000000'9F F989' C2 066A 1189 SUBL R5,R3 ;UNBIAS SVPATE, COULD BE SWAPPED HERE  
 55 53 55 D0 066D 1190 MOVL @#CTL\$GL\_PHD,R5 ;USE P1 SPACE HEADER WINDOW  
 F989' 30 0674 1191 BSBW MMG\$SCNWSLX ;SCAN FOR THE WORKING SET LIST INDEX  
 0677 1192 :  
 0677 1193 : COULD HAVE BEEN SWAPPED IN THE ABOVE ROUTINE, BUT IPL IS BACK AT SYNCH NOW  
 0677 1194 :  
 55 6C A4 D0 0677 1195 MOVL PCBSL\_PHD(R4),R5 ;RECOVER SYSTEM ADDRESS OF PHD  
 53 55 C0 067B 1196 ADDL R5,R3 ;REBIAS PTE ADDRESS  
 51 D5 067E 1197 TSTL R1 ;SEE IF FOUND WORKING SET LIST ENTRY  
 3E 13 0680 1198 BEQL 40\$: ;IF NOT, PAGE WENT AWAY OR SOME OTHER  
 52 6541 D0 0682 1200 20\$: MOVL (R5)[R1],R2 ;PROCESS WAS THE ORIGINATOR OF THE I/O  
 OC BB 0686 1201 PUSHR #^M<R2,R3> ;FETCH WORKING SET LIST ENTRY  
 028A 30 0688 1202 BSBW MMGSFR\$WSLX ;SAVE VIRTUAL ADDRESS AND PTE ADDRESS  
 OC BA 0688 1203 POPR #^M<R2,R3> ;FREE THIS WORKING SET LIST ENTRY  
 04 50 E8 068D 1204 BLBS R0,30\$: ;RECOVER VA AND SVPATE  
 0690 1205 : ;BRANCH IF SUCCESSFUL  
 0690 1206 :  
 0690 1207 : FREWSLX CAN ONLY FAIL IF PAGE FILE NEEDED TO BE ALLOCATED AND IT COULDN'T BE  
 0690 1208 : THIS CASE IS NOT POSSIBLE HERE.

02 02 18 ED 0690 1209 BUG\_CHECK FREWSLX,FATAL  
 FFFF 0690 :WORD "XFEFF  
 0004' 0692 :IIF IDN <FATAL>,<FATAL> , .WORD BUGS\_FREWSLX!4  
 0694 1210 :  
 0694 1211 : IF THIS PAGE FAULT IS FROM USER OR SUPER MODE THEN ISSUE A  
 0694 1212 : PAGE READ ERROR EXCEPTION.  
 0694 1213 :  
 02 24 AE 0694 1214 30\$: CMPZV #PSL\$V CURMOD,#PSL\$S CURMOD,- :IF FAULTING MODE IS  
 24 18 0697 1215 FLTPSLTSP),#PSLSC\_SUPER :USER OR SUPER  
 069A 1216 BGEQ 40\$ :THEN PAGE READ ERROR EXCEPTION  
 069C 1217 :  
 069C 1218 : THIS IS A BAD SITUATION NOW, AN EXCEPTION IN EXEC OR KERNEL MODE WILL  
 069C 1219 : CRASH THE SYSTEM. IF THIS PAGE IS OWNED BY USER OR SUPER THEN TRY  
 069C 1220 : SUBSTITUTING A PAGE OF ZEROS. THIS SHOULD SATISFY THE SYSTEM CODE WHICH  
 069C 1221 : IS ACCESSING THE PROCESS PAGE SINCE IT IS PARANOID ABOUT USER SUPPLIED  
 069C 1222 : DATA. THE NEW PAGE WILL BE EXEC READ WRITE BUT OWNED BY THE ORIGINAL  
 069C 1223 : OWNER. THIS WILL RESULT IN AN ACCESS VIOLATION WHEN THE PAGE IS TOUCHED  
 069C 1224 : IN USER OR SUPER MODE.  
 069C 1225 :  
 50 63 02 17 EF 069C 1226 EXTZV #PTE\$V OWN,#PTESS\_OWN,(R3),R0 ;GET THE PAGE OWNER  
 02 50 D1 06A1 1227 CMPL R0,#PSLSC\_SUPER ;OWNED BY USER OR SUPER?  
 1A 19 06A4 1228 BLSS 40\$ :BRANCH IF NOT, READ ERROR FOR  
 06A6 1229 :A CRUCIAL PAGE, ISSUE THE PAGE  
 06A6 1230 :READ ERROR EXCEPTION, DOWN WE GO.  
 06A6 1231 SETIPL #IPL\$\_ASTDEL  
 12 02 DA 06A6 MTPR #IPL\$\_ASTDEL,S^#PRS\_IPL :LOWEST POSSIBLE FAULT IPL  
 06A9 1232 :  
 06A9 1233 : FORM ARGUMENT LIST FOR CRETVA  
 06A9 1234 :  
 52 DD 06A9 1235 PUSHL R2 :VIRTUAL ADDRESS TO CREATE  
 05 BB 06AB 1236 PUSHR #^M<R0,R2> :ANOTHER COPY OF ADR TO FORM RANGE  
 06AD 1237 :ACCESS MODE PARAMETER  
 01 AE 05 90 06AD 1238 MOVB S^#PRTSC\_EW,1(SP) :SET DESIRED PAGE PROTECTION  
 00 DD 06B1 1239 PUSHL #0 :NULL RETURN ADDRESS  
 08 AE 08 DF 06B3 1240 PUSHAL 8(SP) :ADDRESS OF RANGE TO CREATE  
 00000000'GF 05 FB 06B6 1241 CALLS #5,G^MMGSCRETVA :KERNEL MODE ENTRY TO CRETVA  
 06BD 1242 :PRESERVES IPL  
 06BD 1243 :STRIP OFF INPUT RANGE WHEN DONE  
 FF46 31 06BD 1244 BRW PGFCOMPLETE :FAULT THIS PAGE FROM SCRATCH  
 1A AE B6 06C0 1245 40\$: INCW FLTCTL+2(SP) :INDICATE PAGE READ ERROR  
 06C3 1246 ACVIOLAT:  
 50 26 AE 3C 06C3 1247 MOVZWL FLTPSL+2(SP),R0 :GET IPL FROM FAULT PSL  
 06C7 1248 ENBINT R0 :AND RESTORE IT  
 12 50 DA 06C7 MTPR R0,S^#PRS\_IPL :RESTORE REGISTERS SAVED BY PAGE FAULT  
 03 6E 3F BA 06CA 1249 POPR #^M<R0,R1,R2,R3,R4,R5> :BRANCH IF ACCESS VIOLATION  
 F92D' 10 E5 06CC 1250 BBCC #16,(SP),10\$ :ISSUE THE EXCEPTION  
 F92A' 31 06D0 1251 BRW EXE\$PAGRDR :ACCESS VIOLATION  
 F92A' 31 06D3 1252 10\$: BRW EXE\$ACVIOLAT

06D6 1254 .SBTTL SCANDEADPT - SCAN A DEAD PAGE TABLE FOR TRANSITION PAGES  
 06D6 1255 :  
 06D6 1256 : INPUTS:  
 06D6 1257 :  
 06D6 1258 : R2 = VIRTUAL ADDRESS OF PAGE TABLE (LOW BITS = PAGE TYPE)  
 06D6 1259 : R5 = PROCESS HEADER ADDRESS  
 06D6 1260 : IPL = SYNCH  
 06D6 1261 :  
 06D6 1262 : OUTPUTS:  
 06D6 1263 :  
 06D6 1264 : NONE  
 06D6 1265 :  
 06D6 1266 SCANDEADPT:  
 50 70 A5 6E A5 A3 06D6 1267 SUBW3 PHDSW\_PTCNTVAL(R5),PHDSW\_PTCNTACT(R5),R0 ;ACTIVE PAGE TABLES  
 06DC 1268 : THAT DON'T CONTAIN VALID WSLE'S  
 06DC 1269 : ARE "DEAD PAGE TABLES"  
 03 14 06DC 1270 BGTR 10\$  
 0085 31 06DE 1271 BRW 50\$  
 50 6C A5 A0 06E1 1272 10\$: ADDW PHDSW\_PTCNTLCK(R5),R0 ;BRANCH IF NO DEAD PAGE TABLES  
 50 74 A5 A0 06E5 1273 ADDW PHDSW\_WSFLUID(R5),R0 ;ADD IN THE LOCKED PAGE TABLES  
 50 74 A5 A0 06E9 1274 ADDW PHDSW\_WSFLUID(R5),R0 ;NEED TWICE FLUID EXTRA  
 50 50 3C 06ED 1275 MOVZWL R0,R0 ;GET IT IN A LONGWORD  
 7E 50 A5 3C 06F0 1276 MOVZWL PHDSW\_WSSIZE(R5),-(SP) ;GET CURRENT WS SIZE  
 50 6E C2 06F4 1277 SUBL (SP),R0 ;SUBTRACT OUT CURRENT WSL SIZE  
 6E 0E A5 08 A5 A3 06F7 1278 SUBW3 PHDSW\_WSLIST(R5),PHDSW\_WSDYN(R5),(SP) ;GET LOCKED PORTION OF WS  
 6E 6E 3C 06FD 1279 MOVZWL (SP),TSP) ;GET IT IN A LONGWORD  
 50 8E C0 0700 1280 ADDL (SP)+,R0 ;ADD IN LOCKED ENTRY COUNT  
 61 19 0703 1281 BLSS 50\$ ;BRANCH IF SAFE TO POSTPONE DEAD PAGE  
 0705 1282 : TABLE SCAN  
 50 52 00C8 C5 C3 0705 1283 SUBL3 PHDSL\_POBR(R5),R2,R0 ;BYTE OFFSET FROM FIRST P0 PAGE TABLE  
 59 19 070B 1284 BLSS 50\$ ;BRANCH IF PROCESS HEADER PAGE  
 50 50 F7 8F 78 070D 1285 ASHL #9,R0,R0 ;FORM PAGE NUMBER  
 50 50 55 C0 0712 1286 ADDL R5,R0 ;ADD IN PHD BASE  
 50 68 A5 C0 0715 1287 ADDL PHDSL\_PTWSLEVAL(R5),R0 ;ADD IN OFFSET TO BYTE ARRAY OF COUNTS  
 0719 1288 : OF VALID WSLE'S IN EACH PAGE TABLE  
 60 95 0719 1289 TSTB (R0) ;IS THIS A DEAD PAGE TABLE  
 49 18 071B 1290 BGEQ 50\$ ;BRANCH IF NOT  
 071D 1291 :  
 071D 1292 : R1,R2,R3 ARE PRESERVED UP TO THIS POINT  
 071D 1293 :  
 10 A5 51 B0 071D 1294 MOVW R1,PHDSW\_WSNEXT(R5) ;UPDATE NEXT POINTER  
 0038'CF D6 0721 1295 INCL W^PMSSGL-DPTSCN ;COUNT THESE SCANS  
 50 52 15 09 EF 0725 1296 EXTZV #VASV\_VPN,#VASS\_VPN,R2,R0 ;PAGE NUMBER OF THE PT IN SYSTEM SPACE  
 50 7FE00000 8F CB 072A 1297 BICL3 #^C<PTESM\_VALID!\_PTESM\_PFN>,- ;GET PFN AND VALID BIT  
 50 0000'DF40 00 E5 0730 1298 DW^MMGSGL\_SPTBASE[R0],R0 ;FROM SPT ENTRY FOR THE PAGE TABLE  
 29 50 1F DD 0739 1300 BBCC #PTESV\_VA[ID,R0,40\$] ;CLEAR VALID BRANCH IF IT WAS CLEAR  
 073B 1301 PUSHL #0 ;FLAG FOR MODIFIED PAGE WRITER NEEDED  
 073B 1302 MOVZWL PFN\_REFERENCE -  
 073B 1303 <DW=PFNSAx\_SHRCNT[R0],-(SP)>,- ;NUMBER OF TRANSITION PAGES  
 073B 1304 LONG\_OPCODE=MOVL,-  
 073B IMAGE=SYS\_NONPAGED  
 00000006 .SAVE\_PSECT LOCAL\_BLOCK  
 0000073B' 0006 .PSECT Z\$INITSPFN\_FIXUP\_TABLE  
 3C 000A .ADDRESS ...\_PFN  
 DD 000B .BYTE OPS\_MOVZWL  
 0000073B .BYTE OPS\_MOVL  
 .RESTORE\_PSECT

7E 0000'DF40 3C 073B MOVZWL @W^PFNSAX\_SHRCNT[R0],-(SP)  
 1F 13 0741 1305 BEQL 40\$ :IF NONE, INCONSISTENT  
 52 01FF 8F AA 0743 1306 BICW #VASM\_BYTE,R2 :START SCANNING PT AT BEGINNING  
 53 80 8F 9A 0748 1307 MOVZBL #128,R3 :AT MOST 128 PTE'S  
 50 82 7BA00000 8F CB 074C 1308 20\$: BICL3 #^C<PTESM VALID ! -  
 0754 1309 PTESM\_TYPT ! PTE\$M\_TYPO  
 13 0754 1310 PTESM\_PFN>, (R2)+,R0  
 51 50 EA 07 0754 1311 BEQL 30\$ :GET THE VALID BIT  
 8F 78 0756 1312 ASHL #-PTESV\_TYPO,R0,R1 :AND THE PFN FROM THE PTE  
 0D 13 075B 1313 BEQL 60\$ :BRANCH IF DEMAND ZERO PAGE  
 EC 53 F5 075D 1314 30\$: SOBGTR R3,20\$ :VALID, TYP1, TYP0 ALL 0 IF TRANSITION  
 36 11 0760 1315 BRB 100\$ :BRANCH IF TRANSITION PAGE  
 FFFF 0762 1316 40\$: BUG\_CHECK SCANDEADPT,FATAL :LOOP THROUGH THE PAGE TABLE  
 0004' 0764 .WORD ^XFEFF :ALL DONE, CNT=# I/O REQ OUTSTANDING  
 0766 1317 .WORD .IIF IDN <FATAL>,<FATAL> , .WORD BUGS\_SCANDEADPT!4  
 50 01 D0 0766 1318 50\$: MOVL #1,R0 :SET CONTINUE RATHER THAN RESTART  
 05 0769 1319 RSB :SHRCNT FOR PAGE TABLE IS 0  
 076A 1320 :DIDN'T FIND SHRCNT TRANSITION PAGES  
 076A 1321 :BEFORE RUNNING OFF THE END OF THE PT  
 076A 1322 :  
 076A 1323 : THIS IS A TRANSITION PAGE  
 076A 1324 :  
 076A 1325 :  
 52 03 0C BB 076A 1326 60\$: PUSHR #^M<R2,R3> :SAVE THESE REGISTERS  
 0000'DF40 00 EF 076C 1327 EXTZV #PFNSV\_PAGTYP,#PFN\$S\_PAGTYP,- :GET PAGE LOCATION  
 01 52 D1 0774 1328 @W^PFNSAB\_STATE[R0],R2 :FROM THE STATE BYTE  
 1A 14 0777 1329 CMPL R2,#PFNSC\_MFYPAGLST :ON MODIFIED OR FREE PAGE LIST  
 0E 13 0779 1330 BGTR 90\$ :BRANCH IF NOT ON EITHER  
 077B 1331 BEQL 80\$ :BRANCH IF ON MODIFIED PAGE LIST  
 077B 1332 :  
 077B 1333 : PAGE IS ON THE FREE PAGE LIST  
 077B 1334 :  
 0000'DF40 F882' 30 077B 1335 BSBW MMGSREMPFN :ON FREE LIST, REMOVE IT  
 10 88 077E 1336 BISB #PFNSM\_DELCON,@W^PFNSAB\_STATE[R0] :FORCE DELETE CONTENTS  
 F879' 30 0784 1337 BSBW MMGSRE[PFN] :AND RELEASE THE PAGE  
 0A 11 0787 1338 BRB 90\$ :  
 0789 1339 :  
 0789 1340 : PAGE IS ON MODIFIED PAGE LIST  
 0789 1341 :  
 0000'DF40 10 88 0789 1342 80\$: BISB #PFNSM\_DELCON,@W^PFNSAB\_STATE[R0] :DELETE CONTENTS AFTER WRITING  
 OC AE 01 88 078F 1343 BISB #1,12(5P) :FLAG MODIFIED PAGE RELEASE NEEDED  
 C5 6E BA 0793 1344 90\$: POPR #^M<R2,R3> :RESTORE SAVE REGISTERS  
 03 BA 0795 1345 SOBGTR (SP),30\$ :COUNT DOWN THE TRANSITION COUNT  
 0E 51 E9 079A 1346 100\$: POPR #^M<R0,R1> :CLEAN OFF THE EXHAUSTED COUNT AND FLAG  
 0000'CF D4 079D 1348 BLBC R1,110\$ :BRANCH IF NO MODIFIED PAGE WRITING  
 0000'CF B4 07A1 1349 CLRL W^SCHSGL\_MFYLOLIM :MAKE SURE IT'S WRITTEN SOON  
 51 0B 3C 07A5 1350 CLRW W^SCHSGL\_MFYLIM :CLEAR COUNT, NOT WRITE REQUESTED FLAG  
 5E 04 C0 07A8 1351 MOVZWL #RSNS\_MP[EMPTY,R1] :SET RESOURCE TO WAIT FOR  
 50 D4 07AB 1352 110\$: ADDL #4,SP :RETURN TO ORIGINAL CALLER  
 05 07AD 1353 RSB :SET FAILURE (OR RESTART)

07AE 1355 .SBTTL WSLEPFN - FETCH PFN FROM WORKING SET LIST ENTRY  
 07AE 1356  
 07AE 1357 CALLING SEQUENCE:  
 07AE 1358  
 07AE 1359 BSBW MMG\$WSLEPFN  
 07AE 1360  
 07AE 1361 INPUTS:  
 07AE 1362  
 07AE 1363 R3 = SYSTEM VIRTUAL ADDRESS OF PAGE TABLE ENTRY  
 07AE 1364 FOR A PAGE THAT IS IN THE WORKING SET LIST  
 07AE 1365  
 07AE 1366  
 07AE 1367  
 07AE 1368  
 07AE 1369  
 07AE 1370  
 07AE 1371 OUTPUTS:  
 7B800000 07AE 1372 WSLEPNMSK: R0 = PFN  
 07B2 1373 .LONG ^C<PTESM\_VALID ! PTESM\_TYP0 ! PTESM\_TYP1 ! PTESM\_PGFLVB>  
 07B2 1374 .ENABL LSB  
 07B2 1375  
 07B2 1376 MMG\$WSLEPFN: INPUTS:  
 50 63 F9 AF CB 07B2 1377 BICL3 B^WSLEPNMSK,(R3),R0 ;GET VALID, TYP0, TYP1, PFN/GPTX  
 51 50 EA 8F 78 07B2 1378 ASHL #~PTESV\_TYP0,R0,R1 ;SEE IF TRANSITION OR VALID PAGE  
 06 14 07BC 1379 BGTR FRE\_GBLTRANS ;BRANCH IF NEITHER  
 50 50 15 00 EF 07BE 1380 10\$: EXTZV #PTESV\_PFN,#PTESS\_PFN,R0,R0 ;GET PFN  
 05 07C3 1381 20\$: RSB ;AND RETURN  
 07C4 1382  
 07C4 1383  
 07C4 1384  
 07C4 1385  
 07C4 1386  
 07C4 1387  
 07C4 1388  
 07C4 1389  
 07C4 1390  
 07C4 1391  
 07C4 1392  
 07C4 1393  
 07C4 1394  
 07C4 1395  
 50 0000'DF40 13 51 F5 07C4 1396 SOBGTR R1,WSLVANVAL ;BRANCH IF NOT GLOBAL FORMAT  
 51 50 EA 8F E0 AF CB 07C7 1397 BBCC #PTESV\_TYP0,R0,WSLVANVAL ;CLEAR TYP0, MUST HAVE BEEN SET  
 78 07D3 1398 BICL3 B^WSLEPNMSK,^W^MMG\$GL\_GPTIBASE[R0],R0 ;FETCH MASTER PTE  
 E9 13 07D8 1399 ASHL #~PTESV\_TYP0,R0,R1 ;MAKE SURE THIS IS IN TRANSITION  
 07DA 1400 BEQL 20\$ ;BRANCH IF IT IS, R0 = PFN  
 07DA 1401 WSLVANVAL: BUG\_CHECK WSLVANVAL,FATAL ;WORKING SET LIST ENTRY VIRTUAL  
 FFFF 0004' 07DC 1402 .WORD ^XFEFF ;IIF IDN <FATAL>,<FATAL> .WORD BUG\$ WSLVANVAL!4  
 07DE 1403  
 07DE 1404 .DSABL LSB ;ADDRESS IS NOT VALID

07DE 1406 .SBTTL FREWSLE - FREE A WORKING SET LIST ENTRY  
07DE 1407 :++  
07DE 1408 : FUNCTIONAL DESCRIPTION:  
07DE 1409  
07DE 1410 : THIS ROUTINE CHOOSES A WORKING SET LIST ENTRY, RELEASES THE  
07DE 1411 : PAGE WHICH OCCUPIES IT (IF ANY), MARKS THE ENTRY AVAILABLE, AND  
07DE 1412 : LEAVES THE WSNEXT POINTER POINTING TO THE AVAILABLE ENTRY.  
07DE 1413 : IN RELEASING A PAGE, IF ITS BACKING STORE ADDRESS IS A  
07DE 1414 : "NOT YET ALLOCATED" PAGING FILE ADDRESS, THEN A PAGING FILE VBN  
07DE 1415 : IS ALLOCATED AT THIS TIME. IT IS POSSIBLE THAT NO VBN'S ARE AVAILABLE  
07DE 1416 : AND THUS THIS ROUTINE CAN RETURN UNSUCCESSFULLY.  
07DE 1417  
07DE 1418 : CALLING SEQUENCE:  
07DE 1419  
07DE 1420 BSBW MMGSFREWSLE  
07DE 1421  
07DE 1422 : INPUT PARAMETERS:  
07DE 1423  
07DE 1424 R4 = PCB ADDRESS  
07DE 1425 R5 = PROCESS HEADER ADDRESS - MAY BE P1 SPACE ADDRESS  
07DE 1426 : IF WORKING WITH PROCESS WORKING SET LIST  
07DE 1427 IPL = SYNCH  
07DE 1428  
07DE 1429 : IMPLICIT INPUTS:  
07DE 1430 : NONE  
07DE 1431  
07DE 1432 : OUTPUT PARAMETERS:  
07DE 1433  
07DE 1434 : IF SUCCESSFUL  
07DE 1435 : R0 LOW BIT IS SET  
07DE 1436 : IF NOT SUCCESSFUL  
07DE 1437 : R0 LOW BIT IS CLEAR AND  
07DE 1438 : R1 = RESOURCE TO WAIT FOR (#RSNS\_XXXXX)  
07DE 1439  
07DE 1440 : IMPLICIT OUTPUTS:  
07DE 1441  
07DE 1442 : IF A WORKING SET ENTRY WAS FREED, IT IS PLACED ON THE FREE LIST  
07DE 1443  
07DE 1444 : COMPLETION CODES:  
07DE 1445 : NONE  
07DE 1446  
07DE 1447 : SIDE EFFECTS:  
07DE 1448 : NONE  
07DE 1449  
07DE 1450 :--

07DE 1452 : FOR MMG\$FREWSLX ENTRY POINT

07DE 1453 :

07DE 1454 : INPUTS:

07DE 1455 :

07DE 1456 : R1 = WORKING SET LIST INDEX

07DE 1457 : R2 = VIRTUAL ADDRESS (LOW BITS = PAGE TYPE)

07DE 1458 : R3 = SYSTEM VIRTUAL ADDRESS OF PAGE TABLE ENTRY

07DE 1459 : R4 = PROCESS CONTROL BLOCK ADDRESS

07DE 1460 : R5 = PROCESS HEADER ADDRESS

07DE 1461 : IPL = SYNCH

07DE 1462 :

07DE 1463 : OUTPUTS:

07DE 1464 :

07DE 1465 : R0 = STATUS

07DE 1466 : R1 = RESOURCE TO WAIT FOR IF NOT SUCCESSFUL

07DE 1467 :

07DE 1468 :

07DE 1469 :

07DE 1470 : FOUND AN EMPTY WORKING SET LIST ENTRY, CHECK WHETHER THERE IS A

07DE 1471 : NEW PEAK WORKING SET SIZE AND WHETHER SWAP AREA NEEDS TO GROW.

07DE 1472 :

07DE 1473 : R0 = GPGCNT+PPGCNT

07DE 1474 :

07DE 1475 : .ENABLE LSB

07DE 1476 :

11 36 A5 02 E1 07DE 1477 10\$: BBC #PHDSV\_WSPEAKCHK,PHDSW\_FLAGS(R5),15\$ ;BRANCH IF CANNOT BE  
00000000'GF 50 B1 07E3 1478 :ABOVE PREVIOUS PEAK WORKING SET SIZE

08 1F 07EA 1479 CMPW R0,G^CTL\$GL\_WSPEAK ;ABOVE PREVIOUS RECORDED PEAK?

00000000'GF 50 01 A1 07EC 1480 BLSSU 15\$ ;BRANCH IF NOT

07F4 1481 ADDW3 #1,R0,G^CTL\$GL\_WSPEAK ;YES, NEW PEAK INCLUDES THE PAGE

11 36 A5 04 E1 07F4 1482 :ABOUT TO BE ADDED TO THE WORKING SET

07F9 1483 15\$: BBC #PHDSV\_IWSPEAKCK,PHDSW\_FLAGS(R5),20\$ ;BRANCH IF CANNOT BE

00000000'GF 50 B1 07F9 1484 :ABOVE PREVIOUS PEAK WORKING SET SIZE

08 1F 0800 1485 CMPW R0,G^CTL\$GL\_IWSPEAK ;ABOVE PREVIOUS RECORDED PEAK?

00000000'GF 50 01 A1 0802 1486 BLSSU 20\$ ;BRANCH IF NOT

080A 1487 ADDW3 #1,R0,G^CTL\$GL\_IWSPEAK ;YES, NEW PEAK INCLUDES THE PAGE

10 A5 51 B0 080A 1488 :ABOUT TO BE ADDED TO THE WORKING SET

52 A5 50 B1 080E 1489 20\$: MOVW R1,PHDSW\_WSNEXT(R5)  
2B 1F 0812 1490 :UPDATE NEXT POINTER

CMPW R0,PHDSW\_SWAPSIZE(R5) ;IS THERE ENOUGH ROOM TO SWAP PROCESS?

50 0A A5 08 A5 A3 0814 1491 BLSSU 30\$ ;BRANCH IF YES

50 52 A5 B1 081A 1492 SUBW3 PHDSW\_WSLIST(R5),PHDSW\_WSAUTH(R5),R0 ;GET AUTHORIZED QUOTA

50 1F 1E 081E 1493 CMPW PHDSW\_SWAPSIZE(R5),R0 ;ENOUGH SPACE TO COVER QUOTA?

50 20 A4 D0 0820 1494 BGEQU 30\$ ;BRANCH IF SO, DON'T NEED ANY MORE

19 13 0824 1495 MOVL PCB\$L\_WSSWP(R4),R0 ;GET BLOCK LOCATION OF LAST ALLOCATION

51 52 A5 3C 0826 1496 BEQL 30\$ ;BRANCH IF NON SWAPPING TYPE PROCESS

52 0000'CF 3C 082A 1498 MOVZWL PHDSW\_SWAPSIZE(R5),R1 ;GET CURRENT SIZE

52 51 C0 082F 1499 MOVZWL W^SWP\$GW\_SWPINC,R2 ;NEW INCREMENT

F7CB' 30 0832 1500 ADDL R1,R2 ;NEW DESIRED SIZE

0E 15 0835 1501 BSBW MMGSALLOC\$WPAREA ;ALLOCATE A SWAP AREA, R0-R3 CHANGED

BLEQ 40\$ ;BRANCH IF ALLOCATION FAILED

20 A4 50 D0 0837 1502 MOVL R0,PCB\$L\_WSSWP(R4) ;UPDATE SWAP FILE VBN

52 A5 52 B0 083B 1503 MOVW R2,PHDSW\_SWAPSIZE(R5) ;AND SIZE OF AREA

50 01 3C 083F 1504 30\$: MOVZWL #SSS\_NORMAL,R0 ;SUCCESSFUL RETURN INDICATION

05 0842 1505 RSB

0843 1506

99 11 0843 1507 35\$: BRB 10\$ ;GET BRANCH DESTINATION TO REACH

0845 1508

M 7

```

      D4 0845 1509 40$: CLRL R3 ;FORCE SKIP COUNT TO ZERO
  50 34 A4 36 A4 A1 084B 1510 MOVZWL PHDSW_WSNEXT(R5),R1 ;INDEX TO NEXT CANDIDATE TO DISCARD
  7E 18 A5 08 A5 A3 0851 1511 ADDW3 PCBSW_PPGCNT(R4),PCBSW_GPGCNT(R4),R0 ;CURRENT PAGE COUNT IN USE
      8E 50 B1 0857 1513 SUBW3 PHDSW_WSLIST(R5),PHDSW_WSQUOTA(R5),-(SP) ;GET QUOTA
      65 1A 085A 1514 CMPW R0,(SP)+ ;IS PAGE COUNT WITHIN QUOTA OR EXTENT?
      OE A4 B5 085C 1515 BGTRU 90$ ;BRANCH IF GREATER THAN QUOTA TO FREE
      60 12 085F 1516 TSTW PCBSW_MTXCNT(R4) ;DOES PROCESS HOLD ANY MUTEX?
      51 0A 0861 1517 BNEQ 90$ ;BR IF YES, DO NOT LET IT WAIT
      50 D4 0864 1519 MOVZWL #RSNS_SWPFILE,R1 ;A WSLE, I.E. RUN RATHER THAN WAIT
      05 0866 1520 CLRL R0 ;R1 = RESOURCE TO WAIT FOR
      0867 1521 RSB ;RETURN FAILURE INDICATION
      0867 1522 LCKWSLE_NOTPGTB: ;RETURN RESOURCE TO WAIT FOR IN R1
      0867 1523 BUG_CHECK BADLCKWSLE,FATAL ;LOCKED WORKING SET LIST ENTRY NOT PAGE TABL
      FEFF 0004' 0869 .WORD ^XFEFF ;.IIF IDN <FATAL>,<FATAL> , .WORD BUG$_BADLCKWSLE!4
      086B 1524 WSSIZEERR: ;VALUE OF WSSIZE IS SMALLER THAN IN USE
      FEFF 0004' 086B 1525 BUG_CHECK WSSIZEERR,FATAL ;.WORD ^XFEFF
      086D .IIF IDN <FATAL>,<FATAL> , .WORD BUG$_WSSIZEERR!4
      086F 1526 WSLENOVAL: ;WSL ENTRY NOT VALID
      FEFF 0004' 086F 1527 BUG_CHECK WSLENOVAL,FATAL ;.WORD ^XFEFF
      0871 .IIF IDN <FATAL>,<FATAL> , .WORD BUG$_WSLENOVAL!4
      0873 1528 : ;THIS IS A LOCKED PAGE IN THE DYNAMIC PORTION OF THE WORKING SET LIST
      0873 1529 : ;IT MUST BE A PAGE TABLE PAGE, NOTE THE CONDITION CODES ARE STILL SET
      0873 1530 : ;FROM THE FETCH OF THE WORKING SET LIST ENTRY.
      0873 1531 :
      0873 1532 :

  14 52 F2 18 0873 1533 50$: BGEQ LCKWSLE_NOTPGTB ;BRANCH IF THIS IS NOT A PAGE TABLE
  FE5A 01 E0 0875 1534 BBS #WSLSV_PAGTYP,R2,60$ ;BRANCH TO SKIP GLOBAL PAGE TABLE PAGES
      30 0879 1535 BSBW SCANDEADPT ;SCAN THE PAGE TABLE TO SEE IF "DEAD"
      087C 1536
      0E 50 E8 087C 1537 BLBS R0,60$ ;IF SO RID IT OF TRANSITION PAGES.
      087F 1538 MMGSFREWSLE: ;AND NOW GET A FREE WS LIST ENTRY
      53 0000'CF 3C 087F 1539 MOVZWL W$GN$GW_WSLMXSKP,R3 ;MAX NUMBER OF TB VALID ENTRIES TO SKIP
  51 10 A5 3C 0884 1540 MOVZWL PHDSW_WSNEXT(R5),R1 ;INDEX TO NEXT CANDIDATE TO DISCARD
      6541 D5 0888 1541 TSTL (R5)[R1] ;IS THIS ENTRY FREE?
      12 13 088B 1542 BEQL 80$ ;BRANCH IF SO, CHECK FOR TRULY FREE
      51 D6 088D 1543 60$: INCL R1 ;STEP TO NEXT ENTRY
      12 A5 51 B1 088F 1544 CMPW R1,PHDSW_WSLAST(R5) ;AT THE END YET?
      04 1B 0893 1545 BLEQU 70$ ;CONTINUE
      51 0E A5 3C 0895 1546 MOVZWL PHDSW_WSDYN(R5),R1 ;BACK TO THE TOP
      52 6541 D0 0899 1547 70$: MOVL (R5)[R1],R2 ;R2 = VA FROM WSLE
      41 12 089D 1548 BNEQ 120$ ;BRANCH IF ENTRY IN USE
      50 34 A4 36 A4 A1 089F 1549 80$: ADDW3 PCBSW_PPGCNT(R4),PCBSW_GPGCNT(R4),R0 ;CURRENT PAGE COUNT IN USE
      50 A5 50 B1 08A5 1550 CMPW R0,PHDSW_WSSIZE(R5) ;ARE PAGES IN USE = WORKING SET SIZE?
      16 13 08A9 1551 BEQL 90$ ;BRANCH IF SO, NEED TO REPLACE A PAGE
      52 18 A5 08 A5 A3 08AD 1553 BGTRU WSSIZEERR ;BRANCH IF MORE PAGES IN WS THAN IN WSSIZE
      52 50 B1 08B3 1554 SUBW3 PHDSW_WSLIST(R5),PHDSW_WSQUOTA(R5),R2 ;QUOTA NUMBER OF PAGES-1
      8B 1B 08B6 1555 CMPW R0,R2 ;ARE WE WITHIN QUOTA NUMBER OF PAGES?
      0000'CF 0000'CF D1 08B8 1556 BLEQU 35$ ;BRANCH IF SO, ALLOWED ANOTHER PAGE
      82 19 08BF 1557 CMPL W$CH$GL_GROWLIM,W$CH$GL_FREECNT ;ENOUGH FREE PAGES TO EXTEND?
      50 51 01 C3 08C1 1558 90$: BLSS 35$ ;BRANCH IF SO
      51 D6 08C5 1559 100$: SUBL3 #1,R1,R0 ;SAVE INDEX OF LAST NON-ZERO WSLE
      51
      51 INCL R1 ;STEP TO NEXT ENTRY

```

12 A5 51 B1 08C7 1560 CMPW R1 PHDSW\_WSLAST(R5) ;AT THE END YET?  
   OD 1B 08CB 1561 BLEQU 110\$ ;CONTINUE  
 51 0E A5 3C 08CD 1562 MOVZWL PHDSW\_WSDYN(R5),R1 ;BACK TO THE TOP  
 04 0000'CF E8 08D1 1563 BLBS W^MMG\$GB\_FREWFGLGS,110\$ ;IF SWAPPER REQUESTED, DON'T MOVE LAST  
 12 A5 50 B0 08D6 1564 MOVW R0,PHDSW\_WSLAST(R5) ;SHRINK WSLAST BACK OVER 0 WSLE'S  
   50 08DA 1565 ;THIS IS SAFE BECAUSE WS IS FULL  
 52 6541 D0 08DA 1566 110\$: MOVL (R5)[R1],R2 ;R2 = VA FROM WSLE  
   E5 13 08DE 1567 BEQL 100\$ ;BRANCH IF UNUSABLE FREE ENTRY  
   8C 52 E9 08E0 1568 120\$: BLBC R2,WSLENOVAL ;BRANCH IF ENTRY NOT VALID  
   52 05 E0 08E3 1569 BBS #WSL\$V\_WSLOCK,R2,50\$ ;SKIP ENTRY IF IT IS LOCKED  
 08 0000'CF 00' E1 08E7 1570 BBC S^#EXESV\_TBCHK,W^EXESGL\_ ;CONDITION CODES STILL SET FROM LOAD OF R2  
   3F 52 DA 08ED 1572 MTPR R2,#PRS\_TBCHK ;FLAGS,130\$ :BRANCH IF TBCHK NOT ENABLED  
   03 1C 08F0 1573 BVC 130\$ ;TEST FOR VALID IN TB  
   98 53 F4 08F2 1574 SOBGEQ R3,60\$ ;BRANCH IF NO VALID TRANSLATION  
 10 A5 51 B0 08F5 1575 130\$: MOVW R1,PHDSW\_WSNEXT(R5) ;SKIP PAGE UNLESS COUNT EXHAUSTED  
   F704' 30 08F9 1576 BSBW MMG\$SVAPTECHK ;UPDATE NEXT POINTER  
   08FC 1577 ;RETURN R3 = SYS VA OF PAGE TABLE ENTRY  
   08FC 1578 ;OK FOR PROCESS PAGE TABLES AND  
   08FC 1579 ;PROCESS HEADER PAGES WITH PROCESS PCB ADR  
   08FC 1580 ;.DISABLE LSB  
   08FC 1581 ;R1 = WSLX, R2 = VA FROM WSLE, R3 = SVAPTE, R4 = PCB, R5 = PHD  
   08FC 1582 ;  
 15 63 1A E1 08FC 1583 BBC #PTE\$V\_MODIFY,(R3),MMG\$FREWSLX ;BRANCH IF PAGE NOT MODIFIED  
   0900 1584 ;IF ENTRY NOT VALID MODIFY=0 SO BRANCH  
 0000'CF 0000'CF D1 0900 1585 CMPL W^MPWSGL\_WAITLIM,W^SCHSGL\_MFYCNT ;ABOVE WAIT PROCESS THRESHOLD?  
   OC 14 0907 1586 BGTR MMG\$FREWSLX ;BRANCH IF SO  
 06 0000'CF 00' E0 0909 1587 BBS S^#MMG\$V\_NOWAIT,W^MMG\$GB\_FREWFGLGS,MMG\$FREWSLX  
   090F 1588 ;BRANCH IF THIS IS SWAPPER  
 51 0C 3C 090F 1589 MOVZWL #RSNS\_MPWBUSY,R1 ;R1 = RESOURCE TO WAIT FOR  
   50 D4 0912 1590 CLRL R0 ;RETURN FAILURE INDICATION  
   05 0914 1591 RSB ;RETURN RESOURCE TO WAIT FOR IN R1  
   0915 1592 ;  
   0915 1593 ;.ENABLE LSB  
   0915 1594 ;  
 51 DD 0915 1595 MMG\$FREWSLX:: ;  
   0915 1596 PUSHL R1 ;SAVE WSLX FOR DELETE BY WSLX  
   0917 1597 ;  
 50 63 7B800000 8F CB 0917 1598 ASSUME PTESV MODIFY EQ PTESV\_TYP1 ;  
   091F 1599 BICL3 #^C<PTESM\_VALID!-PTESM\_TYP1 ;FETCH VALID BIT  
   091F 1600 PTESM\_TYP1 ! PTESM\_TYP0 ;- ;PTE TYPE BITS  
   091F 1601 PTESM\_GPTX>,(R3),R0 ;AND PFN/GPTX FROM PAGE TABLE ENTRY  
 OC 50 1F E4 091F 1602 BBSC #PTESV\_VALID,R0,10\$ ;BRANCH IF PTE VALID  
   0923 1603 ;CLEAR VALID BIT IN R0  
   0923 1604 ;  
   0923 1605 ;PAGE TABLE ENTRY NOT VALID,  
   0923 1606 ;PAGE IN TRANSITION AND READINPROG OR  
   0923 1607 ;GLOBAL PTE POINTING TO TRANSITION PTE  
   0923 1608 ;  
 51 50 EA 8F 78 0923 1609 ASHL #-PTESV\_TYP0,R0,R1 ;IF NEITHER TYP1 OR TYP0 IS SET  
   18 13 0928 1610 BEQL 30\$ ;BRANCH TO RELEASE PAGE  
   092A 1611 ;  
   092A 1612 ;THIS WORKING SET LIST ENTRY POINTED TO A PAGE WITH A PAGE TABLE ENTRY  
   092A 1613 ;WHICH IS NEITHER VALID NOR IN TRANSITION. THIS PAGE MUST BE A GLOBAL  
   092A 1614 ;PAGE ON THE WAY IN TO MEMORY. THE GLOBAL PTE MUST BE IN TRANSITION.  
   092A 1615 ;  
 FE97 30 092A 1616 BSBW FRE\_GBLTRANS ;GET PFN IF PAGE IS GLOBAL TRANSITION

```

13 11 092D 1617      BRB    30$          ;RELEASE ACTIVE PAGE
07 50 1A E5 092F 1618      #PTE$V_MODIFY,R0,20$ :CLR MODIFY BIT IN R0, BR IF CLR
0000'DF40 80 8F 88 0933 1620      BISB   #PFNSM_MODIFY,@W^PFNSAB_STATE[R0] ;RECORD MODIFY BIT
03 A3 84 8F 8A 093A 1621      MPH$INVALIDDHK:: :MULTI-PROCESSING CODE HOOKS IN HERE
093A 1622 20$: BICB   #<PTESM_VALID ! PTE$M_MODIFY>@-24,3(R3) :RESET VALID AND MODIFY
093F 1623      :LEAVING TRANSITION PAGE FOR I/O TO SEE
093F 1624      INVALID R2      :INVALIDATE TRANSLATION BUFFER
3A 52 DA 093F      MTPR   R2,S^#PRS_TBIS
0942 1625      MMG$FRE_TRYSKIP:: :HOOK RETURN LOCATION FOR MP SUPPORT
0942 1626 :
0942 1627      : RELEASE THIS WORKING SET LIST ENTRY
0942 1628 :
51 0000'DF40 D0 0942 1629 30$: MOVL   @W^PFNSAL_BAK[R0],R1 :GET BACKING STORE ADDRESS (VBN)
53 51 15 E0 0948 1630      BBS    #PTE$V_CHRPNT,R1,90$ :BRANCH IF CHECKPOINTABLE
0A 0000'DF40 07 E0 094C 1631 40$: BBS    #PFNSV_MODIFY,@W^PFNSAB_STATE[R0],50$ :BRANCH IF PAGE MODIFIED
51 51 09 78 0953 1632      ASHL   #32-PFNSS_BAK,R1,R1 :DOES PAGE HAVE BACKING STORE?
0B 12 0957 1633      BNEQ   70$      :BRANCH IF YES
0959 1634 :
0959 1635      : NULL PAGE FILE BACKING STORE ADDRESS, AND PAGE IS NOT MODIFIED
0959 1636 :
0959 1637      NULLPGFL_NOMFY:
0959 1638      BUG_CHECK MFYNULPGFL,FATAL
FEFF 0004' 0959      .WORD  ^XFEFF
095B      .IIF IDN <FATAL>,<FATAL> , .WORD      BUG$_MFYNULPGFL!4
095D 1639 :
095D 1640      : NOW THAT WE HAVE A MODIFIED COPY OF THE PAGE IN MEMORY, NO NEED FOR OBSOLETE
095D 1641      : PAGE FILE COPY.
095D 1642 :
53 DD 095D 1643 50$: PUSHL   R3      ;SAVE SVAPTE
F69E' 30 095F 1644      BSBW   MMG$DALCBAKSTORE :RELEASE OLD PAGE FILE BACKING STORE
08 BA 0962 1645      POPR   #^M<R3>      ;RESTORE SVAPTE
0964 1646 :
0964 1647      : PFNSAL_BAK[R0] IS ALL SET UP, R0 = PFN, R2 = VA, R3 = SVAPTE (SLAVE IF GBL)
0964 1648 :
0964 1649 70$: ASSUME PFNSC_PROCESS EQ 0
0964 1650      ASSUME PFNSC_SYSTEM EQ 1
0964 1651      ASSUME PFNSC_GLOBAL EQ 2
0964 1652      ASSUME PFNSC_GBLWRT EQ 3
0964 1653      ASSUME PFNSC_PPGTBL EQ 4
0964 1654      ASSUME PFNSC_GPGTBL EQ 5
0964 1655 :
02 52 03 01 EC 0964 1656      CMPV   #WSL$V_PAGTYP,#WSLSS_PAGTYP,R2,#PFNSC_GLOBAL :GLOBAL PAGE?
6E 1F 0969 1657      BLSSU  130$      :BRANCH IF PROCESS OR SYSTEM
5C 19 096B 1658      BLSS   120$      :BRANCH IF PROCESS OR GLOBAL PAGE TABLE
096D 1659 :
096D 1660      : GLOBAL PAGE - MAKE SLAVE PTE INTO GLOBAL FORMAT
096D 1661 :
51 0000'DF40 0000'CF C3 096D 1662      SUBL3  W^MMG$GL_GPTBASE,@W^PFNSAL_PTE[R0],R1 :BYTE INDEX TO GPTE
51 51 1E 9C 0976 1663      ROTL   #32-2,R1,R1 :GLOBAL PAGE TABLE INDEX
097A 1664      ASSUME PTESV_TYP0 EQ PTE$S_GPTX :TYPO ADJACENT TO GPTX FIELD
63 17 00 51 16 E2 097A 1665      BBSS   #PTESV_TYP0,R1,80$ :SET TYPO BIT FOR GLOBAL FORMAT
097E 1666 80$: INSV   R1,#PTESV_GPTX,#PTESS_GPTX+1,(R3) :STORE GPTX + TYPO IN PTE
0983 1667      :CHANGING FROM TRANSITION TO GLOBAL
01B3 30 0983 1668      BSBW   MMG$DECPTREF :SLAVE PTE NO LONGER LOCKED
0986 1669      DECSHR GTR=150$,- :ONE LESS SHARER, BRANCH IF STILL IN USE
0986 1670      IMAGE_FLAG=SYS_NONPAGED

```

.SAVE PSECT LOCAL\_BLOCK  
.PSECT Z\$INIT\$PFN\_FIXUP\_TABLE  
.ADDRESS ...PFN  
.BYTE OPS\_DECW  
.BYTE OPS\_DECL  
.RESTORE PSECT  
DECW @W^PFNSAX\_SHRCNT[R0]  
BGTR 150\$  
BGEQ 30006\$  
BSBW MMG\$SHRCNTNEG

51 0000'DF40 03 A1 84 8F 00000986 0986 0992 30006\$: MOVL @W^PFNSAL\_PTE[R0],R1 ;GET MASTER PTE ADR  
D0 0992 1671 BICB #<PTESM\_VALID ! PTESM MODIFY>@-24,3(R1) ;FORM TRANSITION PTE  
8A 0998 1672 BRB 130\$ ;GO COUNT ONE LESS WSL REF  
3A 11 099D 099F 1673 1674 : SEE IF OLD BACKING STORE ADDRESS SHOULD BE FORGOTTEN AND BAK REINITIALIZED  
OF 51 16 E0 099F 1676 1677 90\$: BBS #PTE\$V\_TYP0,R1,100\$ ;BRANCH IF SECTION PAGE  
51 1F A5 9A 09A3 1678 MOVZBL PHDSB\_PAGFIL(R5),R1 ;MUST CHECK FOR LARGE PAGING FILE  
51 0000'DF41 D0 09A7 1679 MOVL @W^MMG\$GL\_PAGSWPVC[R1],R1 ;GET ADDR OF PAGE FILE CONTROL BLOCK  
OE 1C A1 15 E0 09AD 1680 BBS #PTE\$V\_CHRPNT,PFL\$L\_MAXVBN(R1),110\$ ;BRANCH IF PART OF LARGE VBN  
07 0000'DF40 07 E1 09B2 1681 100\$: BBC #PFNSV\_MODIFY,@W^PFNSAB\_STATE[R0],110\$ ;BRANCH IF UNMODIFIED PAGE  
0000'DF40 1C A5 D0 09B9 1682 MOVL PHDSL\_PAGFIL(R5),@W^PFNSAL\_BAK[R0] ;RESET BACKING STORE ADDRESS  
51 0000'DF40 FF83 D0 09C0 1683 110\$: MOVL @W^PFNSAL\_BAK[R0],R1 ;GET BACKING STORE ADDRESS  
31 09C6 1684 BRW 40\$ ;CONTINUE  
09C9 1685  
09C9 1686 : PROCESS OR GLOBAL PAGE TABLE  
09C9 1688 1689 120\$: CMPZV #WSL\$V\_PAGTYP,#WSL\$S\_PAGTYP,R2,#PFNSC\_PPGTBL ;PROCESS PAGE TABLE?  
09 12 09CE 1690 BNEQ 130\$ ;BRANCH IF NO  
51 42 A5 3C 09D0 1691 MOVZWL PHDSW\_PHVINDEX(R5),R1 ;PROCESS HEADER VECTOR INDEX  
0000'DF41 B6 09D4 1692 INCW @W^PHV\$GL\_REFCBAS[R1] ;ADD A PROCESS HEADER REFERENCE  
09D9 1693 1694 :WHEN PROCESS PAGE TABLE IS PUT  
09D9 1695 130\$: DECREF EQL=140\$ ;INTO TRANSITION STATE  
09D9 DECW @W^PFNSAW\_REFCNT[R0] ;COUNT ONE LESS WSL REF  
0000'DF40 B7 09D9 1696 :  
19 13 09DE 1697 : OTHER REFERENCES OUTSTANDING, COULD BE DIRECT I/O, PAGING I/O  
03 18 09E0 1698 : IF CURRENT PAGE STATE IS "ACTIVE" CHANGE IT TO "RELEASE PENDING" WHICH  
F61B' 30 09E2 1699 : IF REFaulted WILL BE TRANSFORMED BACK TO ACTIVE.  
09E5 1700 : LEAVE "READ IN PROGRESS" STATE AS IS.  
09E5 1701 :  
07 0000'DF40 03 00 ED 09E5 1702 CMPZV #PFNSV\_LOC,#PFNSS\_LOC,@W^PFNSAB\_STATE[R0],#PFNSC\_ACTIVE  
09ED 1703 :UNLESS STATE IS "ACTIVE"  
0000'DF40 03 00 0D 12 09ED 1704 BNEQ 150\$ ;LEAVE IT AS IT WAS  
09EF 1705 INSV #PFNSC\_RELPEND,#PFNSV\_LOC,#PFNSS\_LOC,@W^PFNSAB\_STATE[R0]  
09F7 1706 :OTHERWISE SET RELEASE-PENDING STATE  
03 11 09F7 1707 BRB 150\$  
09F9 1708 : R0 = PFN, REFCNT = 0, R2 = VA, R3 = SVAPTE  
09F9 1709 :  
09F9 1710 :  
09F9 1711 :

51 F604' 30 09F9 1712 140\$: BSBW MMGSRELPFN :REFCNT = 0, RELEASE PFN  
8E D0 09FC 1713 150\$: MOVL (SP)+,R1 :RECOVER SAVED WSLX  
04 10 09FF 1714 BSBB MMGSDELWSLEX :DELETE WORKING SET LIST ENTRY (BY INDEX)  
50 01 3C 0A01 1715 MOVZWL #SSS\_NORMAL,RO :SUCCESSFUL RETURN INDICATION  
05 0A04 1716 RSB  
0A05 1717  
0A05 1718 .DISABLE LSB

0A05 1720 .SBTTL DELWSLEX - DELETE WORKING SET LIST ENTRY BY INDEX  
0A05 1721 :++  
0A05 1722 : FUNCTIONAL DESCRIPTION:  
0A05 1723 : THIS ROUTINE DELETES THE WORKING SET LIST ENTRY INDEXED BY  
0A05 1724 R1, AND PLACES THE WORKING SET LIST ENTRY ON THE FREE LIST.  
0A05 1725 :  
0A05 1726 : CALLING SEQUENCE:  
0A05 1727 :  
0A05 1728 :  
0A05 1729 : BSBW MMGSDELWSLEX  
0A05 1730 : BSBW MMGSDELWSLEPPG  
0A05 1731 :  
0A05 1732 : INPUT PARAMETERS:  
0A05 1733 :  
0A05 1734 : R1 = WORKING SET LIST INDEX  
0A05 1735 : R2 = VIRTUAL ADDRESS IF ENTERING AT DELWSLEPPG  
0A05 1736 : R4 = PROCESS CONTROL BLOCK ADDRESS  
0A05 1737 : R5 = PROCESS HEADER ADDRESS  
0A05 1738 :  
0A05 1739 : IMPLICIT INPUTS:  
0A05 1740 :  
0A05 1741 :  
0A05 1742 :  
0A05 1743 : OUTPUT PARAMETERS:  
0A05 1744 :  
0A05 1745 : R0 PRESERVED  
0A05 1746 :  
0A05 1747 : IMPLICIT OUTPUTS:  
0A05 1748 :  
0A05 1749 :  
0A05 1750 :  
0A05 1751 : COMPLETION CODES:  
0A05 1752 :  
0A05 1753 :  
0A05 1754 :  
0A05 1755 : SIDE EFFECTS:  
0A05 1756 :  
0A05 1757 :  
0A05 1758 :  
0A05 1759 :--

```

      0A05 1761 .ENABL LSB
      0A05 1762
      0A05 1763 MMGSDELWSLEX:::
      53 52 52 03 6541 DO 0A05 1764 MOVL (R5)[R1],R2 :FETCH WORKING SET LIST ENTRY
      EF 0A09 1765 EXTZV #WSL$V_PAGTYP,#WSL$S_PAGTYP,R2,R3 :GET THE PAGE TYPE
      0AOE 1766 CASE R3 <- :AND DISPATCH ON IT
      0AOE 1767 30$,- :PROCESS PAGE
      0AOE 1768 45$,- :SYSTEM PAGE
      0AOE 1769 10$,- :GLOBAL READ ONLY
      0AOE 1770 10$,- :GLOBAL WRITABLE
      0AOE 1771 45$,- :PROCESS PAGE TABLE
      0AOE 1772 45$,- :GLOBAL PAGE TABLE
      05' 00 53 AF 0AOE CASEW R3,#0,S^#<<30011$-30010$>/2>-1
      0A12
      0017' 0A12 .SIGNED_WORD 30$-30010$
      0019' 0A14 .SIGNED_WORD 45$-30010$
      0010' 0A16 .SIGNED_WORD 10$-30010$
      0010' 0A18 .SIGNED_WORD 10$-30010$
      0019' 0A1A .SIGNED_WORD 45$-30010$
      0019' 0A1C .SIGNED_WORD 45$-30010$
      0A1E 30011$: 1773 BUG_CHECK DELWSLEX,FATAL :BAD PAGE TYPE
      FEFF 0A1E .WORD ^XFEFF
      0004' 0A20 .IIF IDN <FATAL>,<FATAL> , .WORD BUG$_DELWSLEX!4
      0A22 1774 :
      0A22 1775 : GLOBAL PAGE, READ ONLY OR WRITABLE
      0A22 1776 :
      34 0E 10 0A22 1777 10$: BSBB DECVALWSLECNT :DEC VALID WORKING SET LIST ENTRY COUNT
      A4 B7 0A24 1778 DECW PCBSW_GPGCNT(R4) :ONE LESS GLOBAL PAGE IN WORKING SET
      05 11 0A27 1779 BRB 50$ :
      0A29 1780 :
      0A29 1781 : PROCESS PAGE, R2 = VIRTUAL ADDRESS
      0A29 1782 :
      0A29 1783 MMGSDELWSLEPPG:::
      36 07 10 0A29 1784 30$: BSBB DECVALWSLECNT :DECREMENT VALID WORKING SET LIST ENTRY CNT
      A4 B7 0A2B 1785 45$: DECW PCBSW_PPGCNT(R4) :ONE LESS PROCESS PAGE IN WORKING SET
      6541 D4 0A2E 1786 50$: CLRL (R5)[R1] :FREE THE WORKING SET LIST ENTRY
      05 0A31 1787 RSB
      0A32 1788
      0A32 1789 .DSABL LSB
      0A32 1790
      0A32 1791 DECVALWSLECNT:
      53 55 68 A5 C1 0A32 1792 ADDL3 PHDSL_PTWSLEVAL(R5),R5,R3 :BASE ADR OF BYTE ARRAY OF COUNTS OF
      0A37 1793 :VALID WSLE'S IN EACH PAGE TABLE
      52 52 0F 10 EE 0A37 1794 EXTV #VASV_VPN+7,#VASS_VPN+1-7,R2,R2 :BITS 16:30 OF VA SIGN EXTENDED
      05 18 0A3C 1795 BGEQ 10$ :BRANCH IF P0 SPACE
      53 0000'CF C0 0A3E 1796 ADDL W^SGNSGL_PTPAGCNT,R3 :END ADDRESS OF BYTE ARRAY
      6342 97 0A43 1797 10$: DECB (R3)[R2] :ONE LESS VALID WSLE IN THIS PAGE TABLE
      03 18 0A46 1798 BGEQ 20$ :BRANCH IF PT STILL HAS OTHER VALID WSLE'S
      6E A5 B7 0A48 1799 DECW PHDSW_PTCNTVAL(R5) :ONE LESS PT WITH VALID WSLE'S
      05 0A4B 1800 20$: RSB

```

0A4C 1802 .SBTTL ININEWPFN - ALLOCATE AND INIT A NEW PFN  
0A4C 1803 ++  
0A4C 1804 FUNCTIONAL DESCRIPTION:  
0A4C 1805  
0A4C 1806 ALLOCATE A NEW PFN AND INITIALIZE THE PFN DATA BASE FOR IT  
0A4C 1807 AND MAKE A WORKING SET LIST ENTRY.  
0A4C 1808  
0A4C 1809 CALLING SEQUENCE:  
0A4C 1810  
0A4C 1811 BSBW MMG\$ININEWPFN ;ALLOCATE AND INIT NEW PFN  
0A4C 1812  
0A4C 1813 INPUT PARAMETERS:  
0A4C 1814  
0A4C 1815 R2 = FAULT VA (LOW BITS = PAGTYP)  
0A4C 1816 R3 = SVAPTE (SLAVE IF GLOBAL)  
0A4C 1817 R4 = PCB ADDRESS (PROCESS IF GLOBAL)  
0A4C 1818 R5 = PROCESS HEADER ADDRESS (PROCESS IF GLOBAL)  
0A4C 1819  
0A4C 1820 IMPLICIT INPUTS:  
0A4C 1821  
0A4C 1822 NONE  
0A4C 1823  
0A4C 1824 OUTPUT PARAMETERS:  
0A4C 1825  
0A4C 1826 R0 = PFN, OR NEGATIVE IF NONE AVAILABLE  
0A4C 1827  
0A4C 1828 IMPLICIT OUTPUTS:  
0A4C 1829  
0A4C 1830 NONE  
0A4C 1831  
0A4C 1832 COMPLETION CODES:  
0A4C 1833  
0A4C 1834 NONE  
0A4C 1835  
0A4C 1836 SIDE EFFECTS:  
0A4C 1837  
0A4C 1838 NONE  
0A4C 1839  
0A4C 1840 ;--

		0A4C	1842	ININEPFNWAIT:					
	2C	BA	0A4C	1843	POPR #^M<R2,R3,R5>	:RESTORE SAVED REGISTERS			
		05	0A4E	1844	RSB	:AND RETURN NO FREE PAGES INDICATION			
			0A4F	1845					
			0A4F	1846	MMGSININEPFN::				
		55	DD	0A4F	1847	PUSHL R5	:SAVE PHD		
		53	DD	0A51	1848	PUSHL R3	:SAVE SVAPTE		
		52	DD	0A53	1849	PUSHL R2	:SAVE VA		
	F0	50	F5A8'	30	0A55	1850	BSBW MMGSALLOCNFN	:ALLOCATE A NEW PFN	
		52	1F	E0	0A58	1851	BBS #31 R0 ININEPFNWAIT	:BRANCH IF NONE AVAILABLE	
02	52	03	01	EC	0A5F	1852	MOVQ (SP) R2	:R2=VA, R3=SVAPTE	
			10	19	0A64	1853	CMPV #WSL\$V_PAGTYP, #WSLSS_PAGTYP, R2, #PFNSC GLOBAL	;GLOBAL PAGE?	
53	63	16	00	EF	0A66	1854	BLSS 40\$	:BRANCH IF NOT	
	53	0000'DF43		DE	0A6B	1855	EXTZV #PTE\$V_GPTX, #PTE\$S_GPTX, (R3), R3	:GLOBAL PAGE TABLE INDEX	
	55	0000'DF		DE	0A71	1856	MOVAL @W^MMG\$GL_GPTBASE[R3], R3	:SVAPTE OF MASTER	
	03	52	1F	E0	0A76	1857	MOVAL @W^MMG\$GL_SYSPHD, R5	:SYSTEM PROCESS HEADER ADR FOR GLOBAL	
		007E		30	0A7A	1858	BBS #VASV SYSTEM, R2, 50\$	:DON'T COUNT PT REF FOR SYSTEM PAGE	
	0000'DF40	53		DO	0A7D	1860	BSBW MMGSINCPTREF	:LOCK PAGE TABLE ENTRY (NOT FOR SYSTEM)	
					0A83	1861	MOVL R3, @W^PFNSAL_PTE[R0]	:STORE PTE ADDRESS	
	0000'DF40	52	52	1F	9C	0A83	1862	ASSUME PFNSV PAGTYP-EQ 0	
					0A87	1863	ROTL #<32-WSL\$V_PAGTYP>, R2, R2	:POSITION PAGE TYPE FIELD	
				2C	BA	0A8F	1864	BICB3 #^C<PFNSM_PAGTYP>, R2, @W^PFNSAB_TYPE[R0]	:SET PAGE TYPE
					0A91	1865	POPR #^M<R2,R3,R5>	:R2=VA, R3=SVAPTE, R5=PHD	
					0A91	1866			
					0A91	1867			
								: FALL THROUGH TO MMGSMAKEWSLE	

0A91 1869 .SBTTL MAKWSLE - MAKE A WORKING SET LIST ENTRY

0A91 1870 ++  
0A91 1871 FUNCTIONAL DESCRIPTION:

0A91 1873 THIS ROUTINE ENTERS SPECIFIED VIRTUAL ADDRESS INTO THE WORKING  
0A91 1874 SET LIST. IT ASSUMES THAT THERE IS A FREE WORKING SET LIST ENTRY ON THE  
0A91 1875 FREE LIST. IF THE PAGE IS A GLOBAL PAGE THE SLAVE PAGE TABLE ENTRY IS  
0A91 1876 LOCKED AT THIS TIME AND THE SHRCNT AND/OR REFCNT IS INCREMENTED.  
0A91 1877 THIS ROUTINE ALSO KEEPS THE ACTIVE PAGE COUNTERS IN THE PCB (PPGCNT, GPGCNT).

0A91 1878  
0A91 1879  
0A91 1880 CALLING SEQUENCE:

0A91 1881 BSBW MMG\$MAKWSLE

0A91 1882  
0A91 1883 INPUT PARAMETERS:

0A91 1884  
0A91 1885  
0A91 1886 R0 = PAGE FRAME NUMBER  
0A91 1887 R2 = VA (LOW BITS = PAGTYP)  
0A91 1888 R3 = SVAPTE (SLAVE IF GLOBAL)  
0A91 1889 R4 = PCB ADDRESS (PROCESS IF GLOBAL)  
0A91 1890 R5 = PHD ADDRESS (PROCESS IF GLOBAL)  
0A91 1891  
0A91 1892  
0A91 1893  
0A91 1894  
0A91 1895 IMPLICIT INPUTS:  
0A91 1896 FREE WORKING SET LIST CONTAINS AT LEAST ONE ENTRY

0A91 1897 OUTPUT PARAMETERS:

0A91 1898 R0 = PFN PRESERVED

0A91 1899  
0A91 1900 IMPLICIT OUTPUTS:

0A91 1901  
0A91 1902 NONE

0A91 1903  
0A91 1904 COMPLETION CODES:

0A91 1905  
0A91 1906 NONE

0A91 1907  
0A91 1908 SIDE EFFECTS:

0A91 1909  
0A91 1910 NONE

0A91 1911  
0A91 1912 --

```

      51 10 A5 3C 0A91 1914 MMG$MAKESL:::          ;WSLX FOR FREE ENTRY
      51 10 A5 3C 0A91 1915 MOVZWL PHDSW_WSNEXT(R5),R1
      51 10 A5 3C 0A95 1916 ASSUME WSL$V VALID EQ 0
      6541 1A 6541 E8 0A95 1917 BLBS (R5)[R1],20$ ;BRANCH IF ENTRY BUSY, ERROR
      52 01 C9 0A99 1918 BISL3 #WSL$M_VALID,R2,(R5)[R1];STORE NEW WSLE
      51 52 03 01 EF 0A9E 1920
      51 52 03 01 EF 0A9E 1921 EXTZV #WSL$V_PAGTYP,#WSL$S_PAGTYP,R2,R1;EXTRACT THE PAGE TYPE
      51 52 03 01 EF 0AA3 1922 CASE R1 <- ;AND DISPATCH ON IT
      51 52 03 01 EF 0AA3 1923 50$,- ;PROCESS PAGE
      51 52 03 01 EF 0AA3 1924 70$,- ;SYSTEM PAGE
      51 52 03 01 EF 0AA3 1925 30$,- ;GLOBAL READ ONLY
      51 52 03 01 EF 0AA3 1926 30$,- ;GLOBAL WRITABLE
      51 52 03 01 EF 0AA3 1927 70$,- ;PROCESS PAGE TABLE
      51 52 03 01 EF 0AA3 1928 70$,- ;GLOBAL PAGE TABLE
      05' 00 51 AF 0AA3 30012$: CASEW R1,#0,S^#<<30013$-30012$>/2>-1
      05' 00 51 AF 0AA7 30012$: .SIGNED_WORD 50$-30012$
      0022' 0AA7 30012$: .SIGNED_WORD 70$-30012$
      0024' 0AA9 30012$: .SIGNED_WORD 30$-30012$
      0010' 0AAB 30012$: .SIGNED_WORD 30$-30012$
      0010' 0AAD 30012$: .SIGNED_WORD 70$-30012$
      0024' 0AAF 30012$: .SIGNED_WORD 70$-30012$
      0024' 0AB1 30012$: .SIGNED_WORD 70$-30012$
      FEFF' 0AB3 30013$: 1929 20$: BUG_CHECK MAKESL,FATAL ;BAD PAGE TYPE OR
      0004' 0AB3 30013$: .WORD ^XFEFF
      0AB5 30013$: .IIF IDN <FATAL>,<FATAL>,.WORD BUG$ MAKESL!4
      0AB7 1930 30013$: ;WSNEXT POINTS TO VA[ID WSLE
      0AB7 1931 : GLOBAL PAGE, READ ONLY OR WRITABLE
      0AB7 1932 :
      0AB7 1933 :
      42 10 0AB7 1934 30$: BSBB MMG$INCPTREF ;LOCK THE SLAVE PAGE TABLE ENTRY
      34 A4 B6 20 10 0AB9 1935 BSBB INCVALWSLECNT ;INC VALID WORKING SET LIST ENTRY COUNT
      34 A4 B6 0ABB 1936 INCW PCB$W PPGCNT(R4) ;ANOTHER GLOBAL PAGE IN WORKING SET LIST
      0ABE 1937 PFN_REFERENCE -
      0ABE 1938 ACBW <#1,#1,aw^PFNSAx_SHRCNT[R0],80$>,- ;COUNT SHARER, BRANCH IF FIR
      0ABE 1939 LONG OPCODE=ACBL,-
      0ABE 1940 IMAGE=SYS NONPAGED
      0ABE 000000012 .SAVE PSECT LOCAL_BLOCK
      000000ABE 0012 .PSECT Z$INIT$PFN_FIXUP_TABLE
      3D 0016 .ADDRESS ...PFN
      F1 0017 .BYTE OPS_ACBW
      000000ABE 0017 .BYTE OPS_ACBL
      000000ABE 0018 .RESTORE PSECT
      000E 0000'DF40 01 01 3D 0ABE 1941 ACBW #1,#1,aw^PFNSAx_SHRCNT[R0],80$ ;INC VALID WORKING SET LIST ENTRY COUNT
      11 11 0AC7 1942 BRB 90$ ;ONE MORE ACTIVE PROCESS PAGE IN WSL
      0AC9 1943 : PROCESS PAGE
      0AC9 1944 :
      36 A4 B6 10 10 0AC9 1945 50$: BSBB INCVALWSLECNT ;INC VALID WORKING SET LIST ENTRY COUNT
      0ACE 1946 70$: INCW PCB$W PPGCNT(R4) ;ONE MORE ACTIVE PROCESS PAGE IN WSL
      0ACE 1947 PFN_REFERENCE -
      0ACE 1948 MOVW <PHDSW_WSNEXT(R5),aw^PFNSAx_WSLX[R0]>,- ;SET INDEX TO WSLE
      0ACE 1949 LONG OPCODE=MOVZWL,-
      0ACE 1950 IMAGE=SYS NONPAGED
      0ACE 000000018 .SAVE PSECT LOCAL_BLOCK
      000000018 .PSECT Z$INIT$PFN_FIXUP_TABLE

```

00000ACE' 0018 .ADDRESS ...PFN  
B0 001C .BYTE OPS MOVW  
3C 001D .BYTE OPS MOVZWL  
00000ACE .RESTORE PSECT  
0000'DF40 10 A5 B0 0ACE MOVW PHDSW\_WSNEXT(R5),@W^PFNSAx WSLX[R0]  
OAD5 1951 ;ONLY FOR PRIVATE PAGES  
OAD5 1952  
0000'DF40 B6 OAD5 1953 80\$: INCW @W^PFNSAW\_REFCNT[R0] ;ANOTHER REFERENCE FOR THE PAGE  
05 OADA 1954 90\$: RSB  
OADB 1955  
OADB 1956 INCVALWSLECNT:  
53 55 68 A5 C1 OADB 1957 ADDL3 PHDSL\_PTWSLEVAL(R5),R5,R3 ;BASE ADR OF BYTE ARRAY OF COUNTS OF  
OAE0 1958 ;VALID WSLE'S IN EACH PAGE TABLE  
52 52 0F 10 EE OAE0 1959 EXTV #VASV\_VPN+7,#VASS\_VPN+1-7,R2,R2 ;BITS 16:30 OF VA SIGN EXTENDED  
05 18 OAE5 1960 BGEQ 10\$ ;BRANCH IF PO SPACE  
53 0000'CF C0 OAE7 1961 ADDL W^SGNSGL\_PTPAGCNT,R3 ;BASE ADR TO NEGATIVE INDEX FROM  
6342 96 OAE2 1962 10\$: INCB (R3)[R2] ;ANOTHER VALID WSLE IN THIS PAGE TABLE  
03 14 OAEF 1963 BGTR 20\$ ;BRANCH IF NOT THE FIRST  
6E A5 B6 OAF1 1964 INCW PHDSW\_PTCNTVAL(R5) ;ANOTHER PAGE TABLE WITH VALID WSLE'S  
05 OAF4 1965 20\$: RSB

0AF5 1967 .SBTTL LOCKPGTB - LOCK PAGE TABLE  
 0AF5 1968 ++  
 0AF5 1969 FUNCTIONAL DESCRIPTION:  
 0AF5 1970  
 0AF5 1971 LOCKPGTB TAKES A VIRTUAL ADDRESS, REFERENCES AND  
 0AF5 1972 LOCKS THE ASSOCIATED PAGE TABLE, AND RETURNS THE SYSTEM  
 0AF5 1973 VIRTUAL ADDRESS OF THE PAGE TABLE ENTRY. IT IS CALLED  
 0AF5 1974 WITH IPL = ASTDEL OR LOWER AND RETURNS WITH IPL = SYNCH.  
 0AF5 1975  
 0AF5 1976 CALLING SEQUENCE:  
 0AF5 1977  
 0AF5 1978 BSBW MMG\$LOCKPGTB  
 0AF5 1979  
 0AF5 1980 INPUT PARAMETERS:  
 0AF5 1981  
 0AF5 1982 R2 = VIRTUAL ADDRESS  
 0AF5 1983 R4 = PROCESS CONTROL BLOCK ADDRESS  
 0AF5 1984 R5 = PROCESS HEADER ADDRESS (P1 SPACE IF PROCESS PCB,  
 0AF5 1985 SYSTEM SPACE OF SYSTEM PCB)  
 0AF5 1986 IPL = ASTDEL OR LOWER  
 0AF5 1987  
 0AF5 1988 IMPLICIT INPUTS:  
 0AF5 1989  
 0AF5 1990  
 0AF5 1991  
 0AF5 1992  
 0AF5 1993  
 0AF5 1994  
 0AF5 1995 R2 PRESERVED  
 0AF5 1996 R3 = SYSTEM VIRTUAL ADDRESS OF PAGE TABLE ENTRY  
 0AF5 1997 IPL = SYNCH  
 0AF5 1998 IMPLICIT OUTPUTS:  
 0AF5 1999  
 0AF5 2000 PAGE TABLE LOCKED VIA INCPTREF  
 0AF5 2001  
 0AF5 2002 COMPLETION CODES:  
 0AF5 2003  
 0AF5 2004  
 0AF5 2005  
 0AF5 2006 SIDE EFFECTS:  
 0AF5 2007  
 0AF5 2008  
 0AF5 2009  
 0AF5 2010  
 0AF5 2011  
 0AF5 2012 MMG\$LOCKPGTB:::  
 F508' 30 0AF5 2013 BSBW MMG\$PTEREF :REFERENCE PTE, GET SVAPTE  
 3A 50 E9 0AF8 2014 :RETURNS AT IPL=SYNCH  
 0AFB 2015 BLBC R0,INCPTREFBUG :BRANCH IF LENGTH VIOLATION  
 0AFB 2016 :  
 0AFB 2017 : FALL THROUGH TO MMG\$INCPTREF  
 0AFB 2018 :

OAFB 2020 .SBTTL INCPTREF - INCREMENT PAGE TABLE REFERENCE COUNT  
OAFB 2021 ++  
OAFB 2022 FUNCTIONAL DESCRIPTION:  
OAFB 2023 THIS ROUTINE ACCEPTS THE ADDRESS OF A PAGE TABLE ENTRY AND  
OAFB 2024 LOCKS THE ASSOCIATED PAGE TABLE INTO MEMORY. IT ALSO MAINTAINS THE  
OAFB 2025 COUNT OF SUCH LOCKED PAGE TABLES IN THE PROCESS HEADER VECTOR.  
OAFB 2026  
OAFB 2027  
OAFB 2028 CALLING SEQUENCE:  
OAFB 2029  
OAFB 2030 BSBW MMG\$INCPTREF  
OAFB 2031  
OAFB 2032 INPUT PARAMETERS:  
OAFB 2033  
OAFB 2034 R3 = SYSTEM VIRTUAL ADDRESS OF PAGE TABLE ENTRY (MASTER IF GLOBAL)  
OAFB 2035 R5 = PROCESS HEADER ADDRESS (SYSTEM IF GLOBAL)  
OAFB 2036  
OAFB 2037 IMPLICIT INPUTS:  
OAFB 2038 NONE  
OAFB 2039  
OAFB 2040  
OAFB 2041 OUTPUT PARAMETERS:  
OAFB 2042 R0,R2,R3 PRESERVED  
OAFB 2043  
OAFB 2044 IMPLICIT OUTPUTS:  
OAFB 2045 NONE  
OAFB 2046  
OAFB 2047  
OAFB 2048  
OAFB 2049 COMPLETION CODES:  
OAFB 2050  
OAFB 2051 NONE  
OAFB 2052  
OAFB 2053 SIDE EFFECTS:  
OAFB 2054  
OAFB 2055 NONE  
OAFB 2056  
OAFB 2057 --

51 53 15 09 EF 0AFB 2059 MMG\$INCPTREF::  
 51 53 15 09 EF 0AFB 2060 EXTZV #VASV\_VPN,#VASS\_VPN,R3,R1 ;GET PAGE NUMBER OF PT CONTAINING THIS PTE  
 51 53 15 09 EF 0AFB 2061 :  
 51 53 15 09 EF 0AFB 2062 : \*\*\*\*\* WARNING \*\*\*\*\* THE FOLLOWING DEPENDS ON GPTBASE = SPTBASE  
 51 0000'DF41 DO 0B00 2063 :  
 51 51 15 00 EF 0B08 2064 MOVL @W^MMGSGL\_SPTBASE[R1],R1 :PTE FOR PAGE TABLE  
 0000'CF 51 D1 0B0D 2065 BGEQ 205 :DISASTER IF NOT VALID  
 09 14 0B12 2066 EXTZV #PTE\$V PFN,#PTE\$S PFN,R1,R1 ;PAGE FRAME NUMBER  
 0B14 2067 CMPL R1,W^MMGSGL\_MAXPFN : IS THERE PFN DATABASE?  
 0B14 2068 BGTR SS : NO, SKIP INCREMENT  
 0B14 2069 PFN\_REFERENCE -  
 0B14 2070 ACBW <#1,#1,@W^PFNSAX\_SHRCNT[R1],10\$>,- ;INC SHRCNT, BRANCH IF FIRST  
 0B14 2071 LONG OPCODE=ACBL,-  
 0B14 2072 IMAGE=SYS NONPAGED  
 0B14 .SAVE PSECT LOCAL\_BLOCK  
 00000001E 00000B14' 001E .PSECT Z\$INIT\$PFN\_FIXUP\_TABLE  
 3D 0022 .ADDRESS ...PFN  
 F1 0023 .BYTE OPS\_ACBW  
 00000B14 .BYTE OPS\_ACBL  
 0001 0000'DF41 01 01 3D 0B14 .RESTORE PSECT  
 05 0B1D ACBW #1,#1,@W^PFNSAX\_SHRCNT[R1],10\$  
 0B1E 2073 SS: RSB  
 0B1E 2074 :  
 0B1E 2075 : SHARE COUNT JUST WENT FROM 0 TO 1 INDICATING THAT THE FIRST ACTIVE  
 0B1E 2076 : PAGE TABLE ENTRY WAS JUST PLACED IN THE PAGE TABLE  
 0B1E 2077 :  
 0B1E 2078 : ASSUMPTION HERE IS THAT THIS ROUTINE IS NOT CALLED FOR SYSTEM PAGE TABLES  
 0B1E 2079 : THIS IS EITHER A PROCESS OR GLOBAL PAGE TABLE.  
 0B1E 2080 :  
 0B1E 2081 10\$: PFN\_REFERENCE -  
 0B1E 2082 MOVZWL <@W^PFNSAX\_WSLX[R1],R1>,- ;WORKING SET LIST INDEX  
 0B1E 2083 LONG OPCODE=MOVZWL,-  
 0B1E 2084 IMAGE=SYS NONPAGED  
 0B1E .SAVE PSECT LOCAL\_BLOCK  
 00000024 00000B1E' 0024 .PSECT Z\$INIT\$PFN\_FIXUP\_TABLE  
 3C 0028 .ADDRESS ...PFN  
 DO 0029 .BYTE OPS\_MOVZWL  
 00000B1E .BYTE OPS\_MOVZWL  
 51 0000'DF41 3C 0B1E .RESTORE PSECT  
 6541 20 C8 0B24 2085 MOVZWL @W^PFNSAX\_WSLX[R1],R1  
 70 A5 B6 0B28 2086 BISL #WSLSM WS[OCK,(R5)][R1] ;SET WORKING SET LOCKDOWN BIT  
 51 42 A5 3C 0B2B 2087 INCW PHDSW\_PTCNTACT(R5) ;ANOTHER ACTIVE PAGE TABLE  
 0000'DF41 B6 0B2F 2088 MOVZWL PHDSW\_PHVINDEX(R5),R1 ;PROCESS HEADER VECTOR INDEX  
 05 0B34 2089 INCW @W^PHV\$GL\_REFCBAS[R1] ;COUNT ANOTHER PAGE TABLE LOCKED  
 0B35 2090 RSB  
 0B35 2091 : PAGE TABLE PAGE WAS NOT VALID  
 0B35 2092 :  
 0B35 2093 20\$: :  
 0B35 2094 INCPTREFBUG:  
 FFFF 0B35 2095 BUG\_CHECK INCPTREF,FATAL ;PAGE TABLE NOT VALID  
 0004' 0B37 .WORD ^XFEFF  
 0B39 2096 .IIF IDN <FATAL>,<FATAL> , .WORD BUG\$ INCPTREF!4  
 :LENGTH VIOLATION FROM LOCKPGTB

0B39 2098 .SBTTL DECPTREF - DECREMENT PAGE TABLE REFERENCE COUNT  
0B39 2099 :++  
0B39 2100 : FUNCTIONAL DESCRIPTION:  
0B39 2101 :  
0B39 2102 : THIS ROUTINE DECREMENTS THE REFERENCE COUNT FOR THE PAGE TABLE  
0B39 2103 : CONTAINING THE PAGE TABLE ENTRY ADDRESSED BY R3. IF THE RESULTING REFERENCE  
0B39 2104 : COUNT INDICATES THAT NO MORE PAGE TABLE ENTRIES ARE IN USE, THE PROCESS  
0B39 2105 : HEADER VECTOR REFERENCE COUNT IS DECREMENTED AS WELL INDICATING A  
0B39 2106 : FREE PAGE TABLE  
0B39 2107 :  
0B39 2108 : CALLING SEQUENCE:  
0B39 2109 :  
0B39 2110 : BSBW MMG\$DECPTREF  
0B39 2111 :  
0B39 2112 : INPUT PARAMETERS:  
0B39 2113 :  
0B39 2114 : R3 = SYSTEM VIRTUAL ADDRESS OF PAGE TABLE ENTRY  
0B39 2115 :  
0B39 2116 : IMPLICIT INPUTS:  
0B39 2117 :  
0B39 2118 : NONE  
0B39 2119 :  
0B39 2120 : OUTPUT PARAMETERS:  
0B39 2121 :  
0B39 2122 : R0,R2,R3 PRESERVED  
0B39 2123 :  
0B39 2124 : IMPLICIT OUTPUTS:  
0B39 2125 :  
0B39 2126 : NONE  
0B39 2127 :  
0B39 2128 : COMPLETION CODES:  
0B39 2129 :  
0B39 2130 : NONE  
0B39 2131 :  
0B39 2132 : SIDE EFFECTS:  
0B39 2133 :  
0B39 2134 : NONE  
0B39 2135 :  
0B39 2136 :--

```

51 53 15 09 0B39 2138 MMG$DECPTREF::          ; INDEX TO SPT ENTRY FOR PAGE TABLE
51 51 0000'DF41 D0 0B3E 2139 EXTZV #VASV VPN,#VASS VPN,R3,R1 ; PTE FOR PAGE TABLE
51 56 18 0B44 2140 MOVL @W^MMGSGL_SPTBASE[R1],R1
51 51 15 00 0B46 2141 BGEQ 40$ ;BRANCH IF NOT VALID, ERROR
0000'CF 51 D1 0B4B 2142 EXTZV #PTESV PFN,#PTESS PFN,R1,R1 ; PAGE FRAME NUMBER FOR PAGE TABLE
49 14 0B50 2143 CMPL R1,W^MMGSGL_MAXPFN ;IS THERE PFN DATA BASE FOR THIS PAGE?
0B52 2144 BGTR 20$ ;NO, SKIP DECREMENT
0B52 2145 DECSHR PFN=R1,GTR=20$,- ;ONE LESS ACTIVE PTE IN THIS PT
0B52 2146 IMAGE_FLAG=SYS NONPAGED

0000002A 00000B52' 002A .SAVE PSECT LOCAL_BLOCK
B7 002E .PSECT Z$INIT$PFN_FIXUP_TABLE
D7 002F .ADDRESS ...PFN
00000B52 0000'DF41 B7 0B52 .BYTE OPS_DECW
42 14 0B57 .BYTE OPS DECL
03 18 0B59 .RESTORE PSECT
F4A2' 30 0B5B DECW @W^PFNSAX_SHRCNT[R1]
BGTR 20$ ;SHARE COUNT JUST WENT TO 0, THIS PAGE TABLE IS NO LONGER REQUIRED
BGEQ 30022$ ;TO REMAIN RESIDENT
BSBW MMG$SHRCNTNEG ;R1 = PFN FOR PAGE TABLE PAGE

0B5E 30022$: ;R1 = PFN FOR PAGE TABLE PAGE

0B5E 2147 : ;R1 = PFN FOR PAGE TABLE PAGE
0B5E 2148 : SHARE COUNT JUST WENT TO 0, THIS PAGE TABLE IS NO LONGER REQUIRED
0B5E 2149 : TO REMAIN RESIDENT
0B5E 2150 : R1 = PFN FOR PAGE TABLE PAGE
0B5E 2151 : ;R1 = PFN FOR PAGE TABLE PAGE

50 DD 0B5E 2152 PUSHL R0 ;SAVE THIS REGISTER
0B60 2153 PFN_REFERENCE -
0B60 2154 MOVZWL <@W^PFNSAX_WSLX[R1],R0>,- ;USE IT TO HOLD THE WORKING SET LIST
0B60 2155 LONG_OPCODE=MOVL,-
0B60 2156 IMAGE=SYS NONPAGED

0B60 00000030 00000B60' 0030 .SAVE PSECT LOCAL_BLOCK
. PSECT Z$INIT$PFN_FIXUP_TABLE
. ADDRESS ...PFN
3C 0034 .BYTE OPS_MOVZWL
D0 0035 .BYTE OPS_MOVL
00000B60 0000'DF41 3C 0B60 .RESTORE PSECT
51 0000'DF DE 0B66 2157 MOVZWL @W^PFNSAX_WSLX[R1],R0 ;ADDRESS OF SYSTEM HEADER
51 53 D1 0B6B 2158 MOVAL @W^MMGSGL_SYSPHD,R1 ;PTE ADR IN SYSTEM HEADER?
51 20 1E 0B6E 2159 CMPL R3,R1 ;BRANCH IF YES, GLOBAL PAGE TABLE
51 53 0000'CF C3 0B70 2160 BGEQU 10$ ;ADR RELATIVE TO BEGIN OF BAL SET
51 51 0000'CF C6 0B76 2161 SUBL3 W^SWPSGL_BALBASE,R3,R1 ;PROCESS HEADER INDEX
51 F7 8F 78 0B7B 2162 DIVL W^SWPSGL_BSLOTSZ,R1 ;DIVIDE BY PAGE SIZE
51 22 10 0B80 2163 ASHL #-9,R1,RT ;DECREMENT PROCESS HEADER REFERENCE COUNT
51 0000'CF C4 0B82 2164 BSBB MMG$DEC$PHDREF1 ;CONVERT PROCESS HEADER INDEX
51 51 09 9C 0B87 2165 MULL W^SWPSGL_BSLOTSZ,R1 ;MULL BY PAGE SIZE
51 0000'CF C0 0B8B 2166 ROTL #9,R1,R1 ;TO PROCESS HEADER ADDRESS
6140 20 CA 0B90 2167 10$: ADDL W^SWPSGL_BALBASE,R1 ;SHUT OFF WORKING SET LOCK
70 A1 B7 0B94 2168 BICL #W$LSM_W$LOCK,(R1)[R0] ;ONE LESS ACTIVE PAGE TABLE
03 19 0B97 2169 DECW PHDSW_PTCNTACT(R1) ;BRANCH IF ERROR
01 BA 0B99 2170 BLSS 40$ ;RESTORE SAVED REGISTER
05 0B9B 2171 20$: POPR #^M<R0> ;AND RETURN

0B9C 2172 : ;PAGE TABLE PTE NOT VALID, OR PAGE TABLE REFERENCE COUNT IS BAD
0B9C 2173 : OR PTCNTACT WENT NEGATIVE
0B9C 2174 : ;PAGE TABLE PTE NOT VALID, OR PAGE TABLE REFERENCE COUNT IS BAD
0B9C 2175 : ;OR PTCNTACT WENT NEGATIVE
0B9C 2176 40$: BUG_CHECK DECPTREF,FATAL ;ERROR IN DECPTREF

```

PAGEFAULT  
V04-000

D 9  
- TRANSLATION NOT VALID EXCEPTION HANDLE 16-SEP-1984 00:43:02 VAX/VMS Macro V04-00  
DECPTREF - DECREMENT PAGE TABLE REFERENC 5-SEP-1984 03:45:49 [SYS.SRC]PAGEFAULT.MAR;1

Page 52  
(25)

FEFF 0B9C  
0004' 0B9E

.WORD "XFEFF  
.IIF IDN <FATAL>, <FATAL> , .WORD BUGS\_DECPTREF!4

PAC  
VO4

OBAD 2178 .SBTTL DECPHDREF - DECREMENT PROCESS HEADER REFERENCE COUNT  
OBAD 2179 ++  
OBAD 2180 : FUNCTIONAL DESCRIPTION:  
OBAD 2181 :  
OBAD 2182 : DECPHDREF REDUCES THE PROCESS HEADER REFERENCE COUNT AND INFORMS  
OBAD 2183 : THE SWAPPER IF THE COUNT GOES TO ZERO. THIS COUNT IS RAISED ONCE  
OBAD 2184 : FOR EACH REASON THAT A GIVEN SPT ENTRY IS BUSY NOT COUNTING THE  
OBAD 2185 : WORKING SET LIST ENTRY REFERENCE. THE FOLLOWING ARE REASONS WHY  
OBAD 2186 : THE REFERENCE COUNT IS INCREASED FOR A GIVEN PAGE TABLE PAGE.  
OBAD 2187 : 1. PLACED ON THE FREE OR MODIFIED LIST  
OBAD 2188 : 2. READ OR WRITE IN PROGRESS  
OBAD 2189 : 3. SHARE COUNT IS ABOVE 0, I.E. IT CONTAINS ACTIVE PTE'S  
OBAD 2190 : THE REFERENCE COUNT IS DECREASED UNDER THE FOLLOWING CONDITIONS:  
OBAD 2191 : 1. SHARE COUNT DECREASED FROM 1 TO 0, I.E. LAST PTE GONE  
OBAD 2192 : 2. READ OR WRITE COMPLETE  
OBAD 2193 : 3. PAGE CONTENTS DELETED (DELCONPFN)  
OBAD 2194 : 4. FAULTED OUT OF TRANSITION STATE  
OBAD 2195 :  
OBAD 2196 : CALLING SEQUENCE:  
OBAD 2197 :  
OBAD 2198 BSBW MMGSDECPHDREF ;R5 = PROCESS HEADER ADDRESS  
OBAD 2199 BSBW MMGSDECPHDREF1 ;R1 = PROCESS HEADER VECTOR INDEX  
OBAD 2200 :  
OBAD 2201 : INPUT PARAMETERS:  
OBAD 2202 : DECPHDREF  
OBAD 2203 : R5 = PROCESS HEADER ADDRESS FOR PAGE TABLE PAGE  
OBAD 2204 :  
OBAD 2205 : DECPHDREF1  
OBAD 2206 : R1 = PROCESS HEADER VECTOR INDEX  
OBAD 2207 :  
OBAD 2208 :  
OBAD 2209 : IMPLICIT INPUTS:  
OBAD 2210 :  
OBAD 2211 : NONE  
OBAD 2212 :  
OBAD 2213 :  
OBAD 2214 : OUTPUT PARAMETERS:  
OBAD 2215 : DECPHDREF  
OBAD 2216 : ONLY R1 ALTERED  
OBAD 2217 :  
OBAD 2218 : DECPHDREF1  
OBAD 2219 : ALL REGISTERS PRESERVED  
OBAD 2220 :  
OBAD 2221 : IMPLICIT OUTPUTS:  
OBAD 2222 :  
OBAD 2223 : NONE  
OBAD 2224 :  
OBAD 2225 : COMPLETION CODES:  
OBAD 2226 :  
OBAD 2227 : NONE  
OBAD 2228 :  
OBAD 2229 : SIDE EFFECTS:  
OBAD 2230 :  
OBAD 2231 : NONE  
OBAD 2232 :  
OBAD 2233 :--  
OBAD 2234 :

```
51 42 A5 3C 0BA0 2235 MMGSDECPHDREF::  
0000'DF41 01 B7 0BA4 2236 MOVZWL PHD$W_PHVINDEX(R5),R1 ;PROCESS HEADER VECTOR INDEX  
05 13 0BA9 2237 MMGSDECPHDREF1::  
F451' 31 0BAB 2238 DECW @W^PHV$GL_REF[BAS[R1]] ;COUNT ONE LESS REFERENCE  
0BAF 2239 BEQL 10$ ;BRANCH IF THAT WAS THE LAST REFERENCE  
OBAC 2240 RSB ;INFORM THE SWAPPER, HEADER MAY GO  
OBAF 2241 10$: BRW SCH$SWP_WAKE ;AND RETURN TO THIS ROUTINE'S CALLER
```

OBAF 2244 .SBTTL INIBLDPKT - INIT FOR CALLING BUILDPKT  
OBAF 2245 ++  
OBAF 2246 FUNCTIONAL DESCRIPTION:  
OBAF 2247  
OBAF 2248 THIS ROUTINE SETS UP R0-R2 FOR A SINGLE PAGE READ/WRITE  
OBAF 2249 TO THE ADDRESS SPECIFIED BY THE BACKING STORE ADDRESS.  
OBAF 2250  
OBAF 2251 CALLING SEQUENCE:  
OBAF 2252 BSBW MMG\$INIBLDPKT  
OBAF 2253  
OBAF 2254 INPUT PARAMETERS:  
OBAF 2255  
OBAF 2256 R2 = BACKING STORE ADDRESS  
OBAF 2257 R3 = PAGE TABLE ENTRY ADDRESS (MASTER IF GLOBAL)  
OBAF 2258 R5 = PROCESS HEADER ADDRESS (SYSTEM HEADER IF GLOBAL PAGE)  
OBAF 2259 THIS IS ONLY USED FOR SECTION TYPE BACKING STORE ADDRESSES  
OBAF 2260 IF THE BACKING STORE ADDRESS IN R2 IS KNOWN TO BE A  
OBAF 2261 PAGING FILE ADDRESS, THEN IT SELF DESCRIBES AND THIS  
OBAF 2262 PARAMETER IS IGNORED.  
OBAF 2263  
OBAF 2264  
OBAF 2265 IMPLICIT INPUTS:  
OBAF 2266  
OBAF 2267  
OBAF 2268  
OBAF 2269  
OBAF 2270  
OBAF 2271 R0 = VIRTUAL BLOCK NUMBER  
OBAF 2272 R1 = SECTION OR PAGE FILE CONTROL BLOCK ADDRESS  
OBAF 2273 R2 = WINDOW ADDRESS  
OBAF 2274 R3 = PAGE TABLE ENTRY ADDRESS (PRESERVED)  
OBAF 2275  
OBAF 2276 IMPLICIT OUTPUTS:  
OBAF 2277  
OBAF 2278  
OBAF 2279  
OBAF 2280 COMPLETION CODES:  
OBAF 2281  
OBAF 2282  
OBAF 2283  
OBAF 2284  
OBAF 2285  
OBAF 2286  
OBAF 2287  
OBAF 2288 SIDE EFFECTS:  
--

```

      21 52 17   OBAF 2290 MMGSINIBLDPKT::          ; MMGSINIBLDPKT
      21 52 16   EO   OBAF 2291 BBS #PFNSV_GBLBAK,R2,10$ ; NOT AN I/O ADDRESS IF GBL BAK
      08 18   EF   OBB3 2292 BBS #PTESV_TYPO,R2,20$ ; BRANCH IF SECTION ADDRESS
  51 51 0000'DF41 DO   OBBC 2293 EXTZV #PFNSV_PGFLX,#PFNSS_PGFLX,R2,R1 ; PAGE FILE INDEX
      16 00   EF   OBC2 2294 MOVL @W^MMG$GL_PAGSWPVC[R1],R1 ; PAGE FILE CONTROL BLOCK ADDRESS
      07 50 15   E1   OBC7 2295 EXTZV #PTESV_PGFLVB,#PTESS_PGFLVB,R2,RO ; PAGE FILE VBN
    7E 1C A1   D2   OBCB 2296 BBC #PTESV_CHKPNTR0,5$ ; SKIP NEXT CHECK IF BIT CLEAR
      50 8E CA   OBCF 2297 MCOML PFLSL_MAXVBN(R1),-(SP) ; SEE IF BIT IS PART OF LARGE VBN OR
      24 12   OBD2 2298 BICL2 (SP)+,RO ; TURN IT OFF IF BIT REPRESENTS CHECKPOINTIN
      2299 5$: BNEQ 40$ ; BRANCH IF GOOD VBN
      2300 OBD4 :
      2301 OBD4 : INVALID BACKING STORE ADDRESS FOR I/O
      2302 OBD4 :
      2303 10$: BUG_CHECK IVBAKADIO,FATAL ; INVALID BACKING STORE ADR FOR I/O
      FEFF 0004' OBD4 .WORD ^XFÉFF
      OBD6 .IIF IDN <FATAL>,<FATAL> , .WORD BUG$_IVBAKADIO!4
      OBD8 2304 :
      OBD8 2305 : SECTION TABLE BACKING STORE ADDRESS
      OBD8 2306 :
      51 55 52 52 32 OBD8 2307 20$: CVTWL R2,R2 ; SECTION TABLE INDEX
      51 51 20 A5 C1 OBD8 2308 ADDL3 PHDSL_PSTBASOFF(R5),R5,R1 ; SECTION TABLE BASE ADDRESS
      6142 DE OBE0 2309 MOVAL (R1)[R2],R1 ; SECTION TABLE ENTRY ADDRESS
  50 53 00C8 C5 C3 OBE4 2310 SUBL3 PHDSL_P0BR(R5),R3,RO ; BYTE OFFSET FROM BASE OF PAGE TABLE
  50 50 FE 8F 78 OBEA 2311 ASHL #-2,RO,RO ; LONG WORD INDEX FROM PAGTBL BASE
  52 08 A1 16 00 EF OBEF 2312 EXTZV #SEC$V_VPX,#SEC$S_VPX,SEC$L_VPXPFC(R1),R2 ; VIRTUAL PAGE NUMBER
      50 52 C2 OBF5 2313 SUBL R2,RO ; RELATIVE PAGE IN SECTION
      OBF8 2314 40$: ASSUME SEC$L_VBN EQ PFLSL_VBN
      OBF8 2315 ASSUME SEC$L_WINDOW EQ PF$L_WINDOW
  50 10 A1 C0 OBF8 2316 ADDL SEC$L_VBN(R1),RO ; FORM FILE VBN
  52 0C A1 D0 OBF8 2317 MOVL SEC$L_WINDOW(R1),R2 ; FILE WINDOW
      05 OC00 2318 RSB
      OC01 2319
      OC01 2320
      OC01 2321 .END

```

..PFN	= 00000B60	R	03	FREEPAGWAIT1	00000398	R	03
ACVIOLAT	000006C3	R	03	FREPAGWAIT 5	000001B8	R	03
AST	00000010			FRE_GBLTRANS	000007C4	R	03
ASTPRM	00000014			GBLBAD	00000150	R	03
BADSYSPAG	00000004	R	03	GBLCRF	0000018F	R	03
BAK	00000024			GBLDZRO	000001CA	R	03
BIT...	= 00000003			GBLDZRO PGFL	00000146	R	03
BUGS_BADLCKWSLE	*****	X	03	GBLNOTRESIDENT	000001D9	R	03
BUGS_DECPTREF	*****	X	03	GBLVALID	00000130	R	03
BUGS_DELWSLEX	*****	X	03	GETPAGELOC	000000BA	R	03
BUGS_FREWSLX	*****	X	03	GET_IRP	000001A5	R	03
BUGS_INCPTREF	*****	X	03	GOT_IRP	00000222	R	03
BUGS_IVBAKADIO	*****	X	03	GPGTBL	0000000F	R	03
BUGS_MAKEWSLE	*****	X	03	GPTX	00000018		
BUGS_MFYNULPGFL	*****	X	03	GPTX_PTE	0000001C		
BUGS_PGFGBLBAD	*****	X	03	INC1	00000028		
BUGS_PGFIPHI	*****	X	03	INC4	0000002C		
BUGS_PGFLOCBAD	*****	X	03	INC512	00000030		
BUGS_SCANDEADPT	*****	X	03	INCPTREFBUG	00000B35	R	03
BUGS_WSLENOVAL	*****	X	03	INCVALWSLECNT	00000ADB	R	03
BUGS_WSLVANVAL	*****	X	03	ININEWPFNWAIT	00000A4C	R	03
BUGS_WSSIZEERR	*****	X	03	IOC\$GL_IRPFL	*****	X	03
CAS_MEASURE	= 00000002			IPL\$_ASTDEL	= 00000002		
CLUSTER	00000020			IPL\$_SYNCH	= 00000008		
CLU_END	000004D9	R	03	IPLHI	00000000	R	03
CLU_END1	000003EB	R	03	IRPSB_PRI	= 00000023		
CLU_END_RESRC	000004F3	R	03	IRPSC_LENGTH	= 000000C4		
CLU_END_RESRC1	000004D7	R	03	IRPSL_AST	= 00000010		
CLU_INI_INC	00000400	R	03	IRPSL_ASTPRM	= 00000014		
CLU_NXT	0000040C	R	03	IRPSW_SIZE	= 00000008		
CLU_SCRATCH_SIZ	0000004C			IRPWAIT_3	00000196	R	03
COUNT	00000021			LCKWSLE_NOTPGTB	00000867	R	03
CTL\$GL_IWSPEAK	*****	X	03	LOCBAD	0000011E	R	03
CTL\$GL_PHD	*****	X	03	MMG\$ALLOCPFN	*****	X	03
CTL\$GL_WSPEAK	*****	X	03	MMG\$ALLOC_SWAREA	*****	X	03
DECVAL\$SLECNT	00000A32	R	03	MMG\$AL_SYSPCB	*****	X	03
DIOCNTWAIT_2	000001A0	R	03	MMG\$CRETVA	*****	X	03
DIR...	= 00000001			MMG\$DAL_CBAKSTORE	*****	X	03
DZRO\$PGWAIT 5	00000528	R	03	MMG\$DEC_PHDREF	00000BA0	RG	03
DZRO_GBL_SEC	00000532	R	03	MMG\$DEC_PHDREF1	00000BA4	RG	03
DZRO_PROC_SEC	000001C7	R	03	MMG\$DECPTREF	00000B39	RG	03
DZRO_PTE_0	0000052E	R	03	MMG\$DELWSLEPPG	00000A29	RG	03
DZRO_PTE_0	00000122	R	03	MMG\$DELWSLEX	00000A05	RG	03
EXES\$ACVIOLAT	*****	X	03	MMG\$FREWSLE	0000087F	RG	03
EXES\$ALONONPAGED	*****	X	03	MMG\$FREWSLX	00000915	RG	03
EXES\$BUILDPKTR	*****	X	03	MMG\$FRE_TRYSKIP	00000942	RG	03
EXES\$DEANONPAGED	*****	X	03	MMG\$GB_FREWFGLS	*****	X	03
EXES\$GL_FLAGS	*****	X	03	MMG\$GL_GPTBASE	*****	X	03
EXES\$PA\$RDERR	*****	X	03	MMG\$GL_MAXGpte	*****	X	03
EXES\$V_NOCLUSTER	*****	X	03	MMG\$GL_MAXPFN	*****	X	03
EXES\$V_TBCHK	*****	X	03	MMG\$GL_PAGSWPVC	*****	X	03
FLTCTC	00000018			MMG\$GL_SPTBASE	*****	X	03
FLTPC	00000020			MMG\$GL_SYSPHD	*****	X	03
FLTPSL	00000024			MMG\$INCPTREF	00000AFB	RG	03
FLTVA	0000001C			MMG\$INI_BLDPKT	00000BAF	RG	03
FP_SAV	0000003C			MMG\$ININEWPFN	00000A4F	RG	03
FREEPAGWAIT	00000377	R	03	MMG\$LOCKPGTB	00000AF5	RG	03

MMG\$MAKEWSLE			PFNSC_PPGTBL	= 00000004
MMG\$PAGEFAULT			PFNSC_PROCESS	= 00000000
MMG\$PGFLTWAIT			PFNSC_RDINPROG	= 00000006
MMG\$PGFLTWAIT_1			PFNSC_RELPEND	= 00000003
MMG\$PTEREF		X 03	PFNSC_SYSTEM	= 00000001
MMG\$REFCNTNEG		X 03	PFNSM_BAK	= 007FFFFF
MMG\$RELPFN		X 03	PFNSM_DELCON	= 00000010
MMG\$REMPFN		X 03	PFNSM_GBLBAK	= 00800000
MMG\$RESRCWAIT		X 03	PFNSM_LOC	= 00000007
MMG\$RLPFNSAVPTE		X 03	PFNSM MODIFY	= 00000080
MMG\$SCNWSLX		X 03	PFNSM_PAGTYP	= 00000007
MMG\$SHRCNTNEG		X 03	PFNSS_BAK	= 00000017
MMG\$SVAPTECHK		X 03	PFNSS_LOC	= 00000003
MMG\$VPCTX		X 03	PFNSS_PAGTYP	= 00000003
MMGSV_NOWAIT		X 03	PFNSV_PGFLX	= 00000008
MMG\$WSLEPFN		X 03	PFNSV_BAK	= 00000000
MPH\$INVALIDHK		X 03	PFNSV_COLLISION	= 00000004
MPW\$GL_WAITLIM		X 03	PFNSV_DELCON	= 00000004
NOTGLOBAL		R 03	PFNSV_GBLBAK	= 00000017
NOTTRANSITION		R 03	PFNSV_LOC	= 00000000
NOTSYSTEM		R 03	PFNSV MODIFY	= 00000007
NULLPGFL_NOMFY		R 03	PFNSV_PAGTYP	= 00000000
OPS_ACBL	= 000000F1		PFNSV_PGFLX	= 00000018
OPS_ACBW	= 0000003D		PFNLIST	= 0000060C R 03
OPS_DECL	= 000000D7		PGFSV_LENVIO	= 00000000
OPS_DECW	= 000000B7		PGFSV_PGTBFILT	= 00000001
OPS_MOVL	= 000000D0		PGFSV_WRTACC	= 00000002
OPS_MOVW	= 000000B0		PGFCOMPLETE	= 00000606 R 03
OPS_MOVZWL	= 0000003C		PGFEXIT	= 00000369 R 03
POADDR	000000B4	R 03	PGFMONITOR	= 000003D R 03
PCBSB_PRIB	= 0000002F		PGFMONITOR1	= 00000A0 R 03
PCBSL_EFWM	= 0000004C		PHDSB_DFPFC	= 00000034
PCBSL_PHD	= 0000006C		PHDSB_PAGFIL	= 0000001F
PCBSL_WSSWP	= 00000020		PHDSB_PGTBPFC	= 00000035
PCBSW_DIOCNT	= 0000003E		PHDSL_POBR	= 000000C8
PCBSW_GPGCNT	= 00000034		PHDSL_P1BR	= 000000D0
PCBSW_MTXCNT	= 0000000E		PHDSL_PAGEFLTS	= 0000004C
PCBSW_PPGCNT	= 00000036		PHDSL_PAGFIL	= 0000001C
PCBSW_STATE	= 0000002C		PHDSL_PGFLTIO	= 00000108
PCB_SAV	= 00000040		PHDSL_PSTBASOFF	= 00000020
PFLSL_PFC	= 0000000B		PHDSL_PTWSLEVAL	= 00000068
PFLSL_MAXVBN	= 0000001C		PHDSV_IWSPEAKCK	= 00000004
PFLSL_VBN	= 00000010		PHDSV_NOACCVIO	= 00000003
PFLSL_WINDOW	= 0000000C		PHDSV_PFMFLG	= 00000000
PFMSM\$ON	***** X 03		PHDSV_WSPEAKCHK	= 00000002
PFNSAB_STATE	***** X 03		PHDSW_FLAGS	= 00000036
PFNSAB_TYPE	***** X 03		PHDSW_PHVINDEX	= 00000042
PFNSAL_BAK	***** X 03		PHDSW_PTCNTACT	= 00000070
PFNSAL_PTE	***** X 03		PHDSW_PTCNTLCK	= 0000006C
PFNSAW_REFCNT	***** X 03		PHDSW_PTCNTVAL	= 0000006E
PFNSAX_SHRCNT	***** X 03		PHDSW_SWAPSIZE	= 00000052
PFNSAX_WSLX	***** X 03		PHDSW_WSAUTH	= 0000000A
PFNSC_ACTIVE	= 00000007		PHDSW_WSDYN	= 0000000E
PFNSC_GBLWRT	= 00000003		PHDSW_WSFLUID	= 00000074
PFNSC_GLOBAL	= 00000002		PHDSW_WSLAST	= 00000012
PFNSC_GPGTBL	= 00000005		PHDSW_WSLIST	= 00000008
PFNSC_MFYPAGLST	= 00000001		PHDSW_WSNEXT	= 00000010

PHDSW_WSQUOTA	= 00000018		RELEASEPEND	0000060F R 03
PHDSW_WSSIZE	= 00000050		RESOURCEWAIT	0000037E R 03
PHVSGC_REFCBAS	***** X 03		RSNS_ASTWAIT	= 00000001
PHVREF_CADR	00000048		RSNS_MPEMPTY	= 0000000B
PMSSAL_TRANSFLT	00000018 RG 02		RSNS_MPWBUSY	= 0000000C
PMSSGL_DPTSCN	00000038 RG 02		RSNS_NPDYNMEM	= 00000003
PMSSGL_DZROFLTS	00000014 RG 02		RSNS_SWPFILE	= 0000000A
PMSSGL_FAULTS	00000000 RG 02		RSRCQAIT_3	= 00000125 R 03
PMSSGL_GVALID	0000003C RG 02		SAVABS...	= 0000004C
PMSSGL_PREADIO	00000008 RG 02		SCANDEADPT	000006D6 R 03
PMSSGL_PREADS	00000004 RG 02		SCH\$GL_CURPCB	***** X 03
PMSSGL_PWRITES	0000000C RG 02		SCH\$GL_FREECNT	***** X 03
PMSSGL_PWRITIO	00000010 RG 02		SCH\$GL_GROWLIM	***** X 03
PMSSGL_RDFLTS	00000004 RG 02		SCH\$GL_MFYCNT	***** X 03
PPGTBL	00000063 R 03		SCH\$GL_MFYLIM	***** X 03
PPGTBL_2	0000012B R 03		SCH\$GL_MFYLOLIM	***** X 03
PRS_IPC	= 00000012		SCH\$GL_RESMASK	***** X 03
PRS_TBCHK	= 0000003F		SCH\$GQ_COLPGWQ	***** X 03
PRS_TBIS	= 0000003A		SCH\$GQ_FPGWQ	***** X 03
PRI	00000023		SCH\$GQ_MWAIT	***** X 03
PROCPAG	00000362 R 03		SCH\$GQ_PFWQ	***** X 03
PRTSC_EW	= 00000005		SCH\$SWP_WAKE	***** X 03
PSLSC_SUPER	= 00000002		SCH\$WAITM	***** X 03
PSLSS_CURMOD	= 00000002		SEC\$B_PFC	= 0000000B
PSLSS_IPL	= 00000005		SEC\$L_VBN	= 00000010
PSL\$V_CURMOD	= 00000018		SEC\$L_VPXPFC	= 00000008
PSL\$V_IPL	= 00000010		SEC\$L_WINDOW	= 0000000C
PTE\$C_KW	= 10000000		SEC\$S_VPX	= 00000016
PTE\$M_DZRO	= 00020000		SEC\$V_VPX	= 00000000
PTE\$M_GPTX	= 003FFFFF		SETSLAVEPTE	00000642 R 03
PTE\$M MODIFY	= 04000000		SGN\$GL_BALSETCT	***** X 03
PTE\$M OWN	= 01800000		SGN\$GL_PTPAGCNT	***** X 03
PTE\$M_PFN	= 001FFFFF		SGN\$GW_WSLMXSKP	***** X 03
PTE\$M_PGFLVB	= 003FFFFF		SIZ...	= 00000001
PTE\$M_PROT	= 78000000		SIZE_TYPE	= 00000008
PTE\$M_TYPO	= 00400000		SS\$ NORMAL	= 00000001
PTE\$M_TYP1	= 04000000		STATE	= 00000022
PTE\$M_VALID	= 80000000		SVAPTE	= 00000004
PTE\$S_GPTX	= 00000016		SWP\$GL_BALBASE	***** X 03
PTE\$S OWN	= 00000002		SWP\$GL_BSLOTSZ	***** X 03
PTE\$S_PFN	= 00000015		SWP\$GW_SWPINC	***** X 03
PTE\$S_PGFLVB	= 00000016		SYSTEM\$SPACE	00000042 R 03
PTE\$V_CHKPNT	= 00000015		TMP...	= 00000001
PTE\$V_CRF	= 00000010		TRANSITION	000000FF R 03
PTE\$V_DZRO	= 00000011		TRY_TO_CLUSTER	000003C3 R 03
PTE\$V_GPTX	= 00000000		VA	= 0000000C
PTE\$V MODIFY	= 0000001A		VASM_BYTE	= 000001FF
PTE\$V OWN	= 00000017		VASS_VPN	= 00000015
PTE\$V_PFN	= 00000000		VASV_P1	= 0000001E
PTE\$V_PGFLVB	= 00000000		VASV_SYSTEM	= 0000001F
PTE\$V_TYPO	= 00000016		VASV_VPN	= 00000009
PTE\$V_TYP1	= 0000001A		VALID	000000B1 R 03
PTE\$V_VALID	= 0000001F		VBN	= 00000034
PTEDAT	00000000		WINDOW	= 00000038
QUEUE_PAGE_READ	0000033E R 03		WQH\$W_WQCNT	= 00000008
READERR	0000065B R 03		WQH\$W_WQSTATE	= 0000000A
READINPROG	00000382 R 03		WRITEINPROG	0000060F R 03

WSL\$C_GLOBAL	= 00000004
WSL\$C_GPGTBL	= 0000000A
WSL\$C_PPGTBL	= 00000008
WSL\$C_SYSTEM	= 00000002
WSLSM_PAGTYP	= 0000000E
WSLSM_VALID	= 00000001
WSLSM_WSLOCK	= 00000020
WSLSS_PAGTYP	= 00000003
WSLSV_PAGTYP	= 00000001
WSLSV_VALID	= 00000000
WSLSV_WSLOCK	= 00000005
WSLENOVAL	0000086F R 03
WSLEPFNMSK	000007AE R 03
WSLVANVAL	000007DA R 03
WSSIZEERR	0000086B R 03

```
+-----+
! Psect synopsis !
+-----+
```

PSECT name	Allocation	PSECT No.	Attributes
ABS .	00000000 ( 0.)	00 ( 0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
SABSS	0000004C ( 76.)	01 ( 1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
SS\$210	00000040 ( 64.)	02 ( 2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG
SMMGSCOD	00000C01 ( 3073.)	03 ( 3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG
Z\$INIT\$PFN_FIXUP_TABLE	00000036 ( 54.)	04 ( 4.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

```
+-----+
! Performance indicators !
+-----+
```

Phase	Page faults	CPU Time	Elapsed Time
Initialization	31	00:00:00.05	00:00:02.27
Command processing	121	00:00:00.51	00:00:07.06
Pass 1	571	00:00:21.08	00:01:14.69
Symbol table sort	0	00:00:02.88	00:00:08.42
Pass 2	414	00:00:06.48	00:00:21.46
Symbol table output	1	00:00:00.24	00:00:00.83
Psect synopsis output	0	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	1140	00:00:31.26	00:01:54.75

The working set limit was 2100 pages.

152632 bytes (299 pages) of virtual memory were used to buffer the intermediate code.

There were 100 pages of symbol table space allocated to hold 1753 non-local and 129 local symbols.

2321 source lines were read in Pass 1, producing 32 object records in Pass 2.

39 pages of virtual memory were used to define 36 macros.

+-----+  
! Macro library statistics !  
+-----+

Macro library name

-----  
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1  
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2  
TOTALS (all libraries)

Macros defined

-----  
20  
13  
33

1851 GETS were required to define 33 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LI\$:\$PAGEFAULT/OBJ=OBJ\$:\$PAGEFAULT MSRC\$:\$PAGEFAULT/UPDATE=(ENH\$:\$PAGEFAULT)+EXECML\$/LIB

0378 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

